

CLUB 80

CLUBINFO

DER

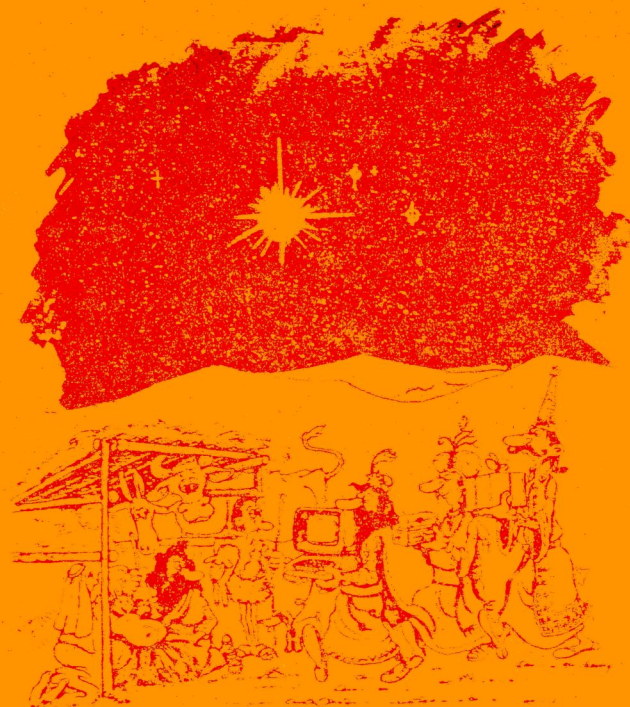
TANDY -

GENIE -

UND KOMTEK -

ANWENDER

16. AUSGABE



Frohes Fest
Frohes Fest 1986
Frohes Fest

KONTAKTADRESSE : CLUB 80 / PETER STEVENS / POSTFACH 56 / 4600 DORTMUND 1
TEL.: 0231 / 593883

Inhaltsverzeichnis

Seite:

Clubintaches

Neues vom Vorstand	81 - 82
Mitgliedsbeitrag	83
Hobbytronik-Treff	83
In eigener Sache	84
Meinung: Wettbewerb	85
Terraine / Messen	86

Software

Programmierwettbewerb	87
Logische Operationen 2	89 - 10
Logiksimulation mit Basic	11 - 13
Logisches zu Logik	14 - 17
BRATIE- Laufende Programme erweitern	18
Spline Interpolation (kubisch)	19 - 20
Goldener FIBONACCI	21 - 23
Steno-Logik	24 - 26
Graphik-JKL, Tscript und MEC PB	27 - 28
Telefon, ein BD-Zeichen	29 - 31
Syskopy und Readisk	32
Mein 3. Assemblerprogramm	33 - 36
RPMI-Programmiersprache	37 - 40

Hardware

Hardwaretip	41 - 44
Taktumschaltung (allerletzte Lösung)	45 - 46

Börse

Wer hat was ???--Wer sucht was ???	47 - 48
Wer kann weiterhelfen	47 - 48

Sonstiges

Der Kniff mit den Logik	46
Be Anfang war ein Hoquetua	46
TANDY - better again	48
Drucker	44
Software zum Nulltarif	48
Bits übers Radio	48

Programmbibliothek

Neues von der DISKO-Front	49 - 51
Diskotheken gesucht	53 - 54

Die letzten Seiten

Impressum	55
Schluss	56
Clubmitgliederadressen	aa INFO-Ende
Sonderteil DIGITAL	aa INFO-Ende
Sonderteil GIIIS	aa INFO-Ende

Neues vom Vorstand

Vorstandssitzung

Am Sonntag den 16.11.86 13:30 Uhr fand in einem Hotel in Höhr-Grenzhausen eine Sitzung des Vorstands statt. Über diese Sitzung will ich euch so ausführlich wie möglich berichten.

Der Vorstand war, bis auf K.J. Mühlenbein, vollzählig. Es nahmen also Teil: 1. Vorsitzender Peter Stevens
2. stellv. Vorsitzender Hartmut Obermann
3. Hardwarekoordinator Eckehard Kuhn
4. Clubinforedakteur Jens Neueder

Die angesetzten Themen waren:

1. Übergabe der Geschäftsführung des CLUB 80 vom ersten an den zweiten Vorsitzenden.
2. Eintragung des CLUB 80 als e.V.
3. Planung des jährlichen Clubtreffens
4. Weitere Clubaktionen

Zu 1.: Da unser 1. Vorsitzender Peter Stevens durch sein ständig expandierendes Geschäft (er ist selbstständig) keine Zeit mehr für die korrekte Abwicklung der Geschäftsführung des CLUB 80 hat, habe ich diese übernommen. Hauptsächlich handelt es sich dabei um die Betreuung der Mitglieder, Aufnahme der Neuzugänge, Verwaltung der Clubbücherei und last not least das Führen der Clubkasse. Die Besetzung des Vorstandes an sich bleibt von dieser Umverteilung der Arbeit unberührt. Unser 1. Vorsitzender bleibt also weiterhin Peter Stevens!

Zu 2.: Inzwischen (es hat ja lange genug gedauert) wurde unsere Satzung vom Amtsgericht als rechtlich einwandfrei anerkannt. Danach könnte nun also die eigentliche Eintragung als e.V. erfolgen. Der Vorstand hat sich jedoch dafür entschieden, mit der Einschreibung noch bis zum nächsten Clubtreffen zu warten, um dann den Club sofort mit dem neuen Vorstand eintragen zu lassen. Ansonsten würde schon Ende März (siehe auch 3.) ein weiterer kostenpflichtiger Gang zum Amtsgericht notwendig!

Zu 3.: Das nächste Clubtreffen des CLUB 80 findet (wenn nichts außergewöhnliches dazwischenkommt) Mitte März statt. Der genaue Termin lautet 13.-15.03.1987! Auf diesen relativ frühen Termin hat sich der Vorstand geeinigt, um nicht mit dem Regionaltreffen "nord" (angeregt und organisiert durch Hans Martin Stephan, Termin: 25./26.04.87) zu konkurrieren. Schließlich soll die Jahreshauptversammlung nicht zu einem Regionaltreffen "süd/mitte" werden, sondern alle Mitglieder ansprechen! Das Treffen wird diesmal etwas weiter nördlich, nämlich im Bereich zwischen Fulda und Kassel (ABK Hattenbach) stattfinden. Wir hoffen damit allen Mitgliederwünschen bezüglich Anfahrtstrecke und Entfernung gerecht zu werden. Der genaue Ort des Treffens steht noch nicht fest, vielmehr habe ich Jürgen Wucherer, der uns schon beim ersten Treffen gut beraten hat, auf die Suche nach einem passenden Hotel geschickt. Im nächsten Info, das Ende Januar bei euch eintrudeln wird, wird dann der Austragungsort bekanntgegeben und ein Anmeldezettel beigelegt von denen hoffentlich recht viele mit positivem Bescheid wieder bei mir landen. Merkt euch also das Datum 13.-15.03.1987 unbedingt vor!!

Zu 4.: Unter Punkt 4 wurden mehrere Unternehmungen besprochen, die der Club und hier vor allem der Vorstand in nächster Zeit durchführen wird.

I. mit dem nächsten Heft zusammen wird ein Gesamt-inhaltsverzeichnis der bisher erschienenen Infos herauskommen.

II. Durch unser Mitglied Gerald Dreyer soll versucht werden, Kontakt zu einem TRS 80-Club in Holland aufzunehmen. Dabei wird vor allem an einen Austausch der Infos und Programmbibliotheken gedacht. Dabei tritt natürlich das Sprachproblem störend in Erscheinung. Sollte jemand die holländische Sprache verstehen, wäre ich dankbar, wenn er sich bei mir meldet. Auch zu einem Club in den USA wollen wir in Verbindung treten.

III. Für unseren Zahnzeiler-Wettbewerb wurden die Preise festgesetzt. Dem Gewinner des ersten Preises wird für das Jahr 1987 der Jahresbeitrag erlassen! Immerhin nur den halben Jahresbeitrag braucht der Gewinner des zweiten Preises zu berappen. Und dem Dritten im Bunde winken als Preis zehn Disketten. Vielleicht wird ja durch diese Preise das Interesse und die Teilnahmefreudigkeit noch etwas angeregt. Einen Einsendeschluß gibt es noch nicht, jeder kann noch mitmachen und hat noch die Chance zu gewinnen!

Soweit also zur Sitzung des Vorstands. Ich hoffe, alle Mitglieder sind mit den Entscheidungen, welche auf dieser Sitzung gefällt wurden einverstanden. Natürlich ist der Vorstand immer offen für Kritik und Anregungen und so würde ich mich freuen, wenn aufgrund dieses Sitzungsberichts Reaktionen (negative oder positive) eurerseits bei mir eintrudeln!

Weihnachts- und Neujahrswünsche 1987!!!

Kaum zu glauben, daß das Jahr 1986 sich schon so rapide seinem Ende zuneigt! Mir kommt es immer noch so vor, als wäre das Clubtreffen von ein paar Wochen gewesen, da steht schon das nächste ins Haus. Sechs Rieseninfos hat der CLUB 80 in diesem Jahr auf die Beine gestellt und dabei ist die Qualität nicht etwa gesunken sondern noch kräftig gestiegen. Das recht langsam angelaufene Hardwareprojekt hat in diesem Jahr nun doch erhebliche Fortschritte gemacht und auch die Softwarebibliothek kann sich sehen lassen!

Ihr seht, ich versuche recht verzweifelt ein Fazit, einen Schlußstrich zu ziehen hinter dieses Jahr. Warum es mir nur schlecht gelingt liegt wohl daran, daß unter eine noch laufende, sich ständig in Bewegung befindende Sache nur schwer ein Strich zu ziehen ist. Wie soll ich dieses Jahr abschließen, wenn ich schon an die Aktivitäten im nächsten denke!?

Ich kann eigentlich nur hoffen, daß sich der Club weiter so entwickelt, wie er es bisher getan hat. Dann brauche ich keinen Nachruf auf 1986 zu schreiben, sondern kann getrost allen Mitgliedern ein frohes Weihnachtsfest und ein glückliches Neues Jahr 1987 wünschen, euer

Hartmut Obermann

P.S.: Besonders herzlich fällt dieser Gruß natürlich an all jene aus, die sich aktiv an der Gestaltung des Clublebens, egal in welcher Form auch immer, beteiligt haben und hoffentlich auch 1987 beteiligen werden!

HEFT
16
November
1986

11

Eine der Aufgaben des geschäftsführenden Vorstands ist das Führen der Clubkasse. Dazu gehört natürlich nicht nur das Ausgeben des Geldes innerhalb des Geschäftsjahres, sondern auch das Eintreiben der Mitgliedsbeiträge. Von meinem Vorgänger Günther Wagner weiß ich, daß dies sogar eine der zeitaufwendigsten Beschäftigungen ist, die einem Vorstand aufgehalst werden. Aus diesem Grunde wurde durch den Vorstand während seiner letzten Sitzung folgende Regelung getroffen:

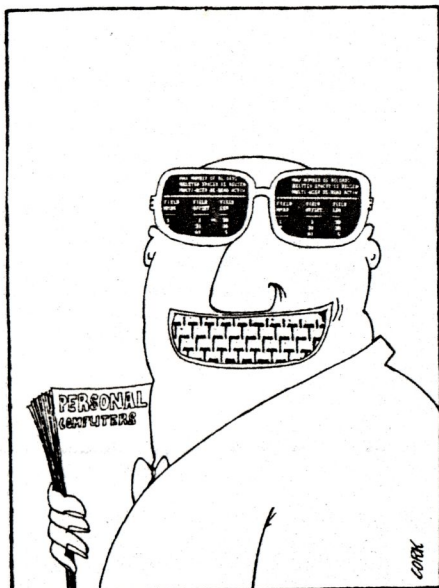
Absolut letzter Termin
für die Einzahlung des Beitrages ist der
15. Januar 1987!

"Absolut letzter Termin" heißt, wer bis dahin seinen fälligen Beitrag nicht bezahlt hat, bekommt das Info Nr. 17, welches Ende Januar erscheint, nicht mehr! Durch diese Maßnahme hofft der Vorstand längeren Mahnaktionen und einem Verlust durch das Versenden von Infos an ohne Abmeldung ausgetretene Mitglieder vorzubeugen.

Ich hoffe ihr versteht und unterstützt diese Maßnahme, euer

Rantaut Obermann

Kontonummer weiterhin: Postgirokonto Peter Stevens
Sonderkonto CLUB 80
Konto-Nummer 285 491 - 465
Postgiroamt Dortmund
BLZ 440 100 46



Hobbytronik-Treff

Wie im letzten Info unter "TERMINE" angekündigt, trafen sich am 19.10. auf der Hobbytronik in Stuttgart Mitglieder des CLUB 80. Dieses Treffen war nicht weiter organisiert, sondern jeder der Lust, Zeit und Geld (letzteres war nicht unbedingt Bedingung, ich bin schließlich auch hingefahren und habe trotzdem kein Geld) hatte konnte sich einfinden. Tatsächlich zusammengefunden haben sich insgesamt 10 Leute (8 Mitglieder + 2 bessere Hälften). Es waren anwesend: Bernd Drohwälder, Ulrich Böckling, Klaus Hermann, Eckehart Kuhn, Jens Neueder mit Freundin Sabine, Richard Rensch mit Sohn und last not least meine Frau mit ihrem ständigen Begleiter.

Ich finde solche lockeren Treffs immer wieder schön und anregend. Einerseits lernt man dabei Mitglieder kennen, die man wohl sonst nie zu Gesicht bekommen hätte, andererseits trifft man immer wieder Freunde mit gleichen Interessen (z.B. war die Gruppe Böckling/Neueder/Obermann schon mehrfach zusammen auf der Hobbytronik, sowohl in Stuttgart als auch in Dortmund) und hat so Gelegenheit zum Meinungsaustausch. Ich würde mich freuen, wenn sich noch mehr Mitglieder bei solchen Kurztreffs sehen ließen und hoffe auch Dich beim nächstenmal zu treffen! Euer Rantaut Obermann

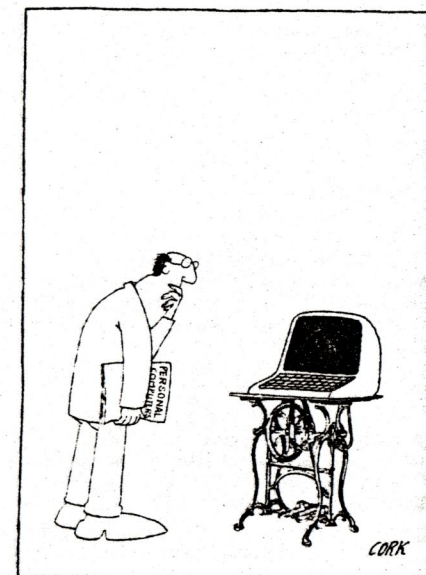
In eigener Sache

Für das kommende INFO zum Jahreswechsel möchte ich eine kleine Themenvorschau geben:

- 1-- Jahresinhaltsverzeichnis aller bisherigen INFO's
- 2-- Komplette Liste der Mitglieder
mit deren Hardwarekonfiguration
(wer dazu noch Änderungen bekanntzugeben hat
sollte dies bitte umgehend tun)
- 3-- Programm-Katalog unserer Diskothek

Punkt 2 hatte ich Euch ja schon länger versprochen. Leider bin ich zwangs Zeitmangel nicht mehr dazu gekommen. Auch sind die Daten noch nicht von allen Mitgliedern vorhanden. Deshalb kann ich auch nur eine unvollständige Liste bieten.

Die Redaktion



Gorbatschow fordert Reagan auf: »Fragen Sie mal Ihren Computer, was mit Amerika im Jahre 2000 sein wird.« Sekunden später kommt die Antwort: »Sozialistisch.« Als Gorbatschow seinen Computer nach der Zukunft der Sowjetunion befragt, sieht er verwirrt auf den Bildschirm und meint: »Kann ich nicht lesen – das ist Chinesisch!«

»Können Sie mir helfen? Ich suche ein Geschenk für meinen Onkel – er wird morgen 64 Jahre alt.«
»Da haben Sie aber Glück! Ich habe hier zufällig einen Schachcomputer mit genau 64 Feldern.«

MEINUNG:

Zum Thema Programm - Wettbewerb...

Der Peter bezeichnete das Preisausschreiben einen 33-Zeiler zu programmieren als Rainfall und es war wohl auch einer. Doch woran mag es gelegen haben, da wir doch einige wahre Spitzenleute in unserem Club beherbergen? (ich gehöre nicht dazu)

Persönlich störte ich mich an der Rahmenbedingung. Denn die Zeilenanzahl ist wohl das letzte worauf beim Programmerstellen geachtet wird!!! Ganz abgesehen davon, daß die Übersicht enorm darunter leidet. Als Beispiel habe ich nochmals mein Interpolationsprogramm ausgedruckt. Bei der 10-Zeilenform blicke ich jetzt nach einem Monat nämlich selber nicht mehr so ganz durch.

Aus diesem Grunde will ich vorschlagen, daß in Zukunft doch besser ein konkretes Problem vorgelegt wird, wobei sinnvolle Nebenbedingungen...

hohe Geschwindigkeit
geringer Speicherbedarf
anwenderfreundlich

(zum Beispiel) im Vordergrund stehen. Übersichtlichkeit und eine Minidokumentation im Programm sollten ja selbstverständlich sein. Die Sprache (Assembler, Basic, C, Cobol, Fortran, Pascal...) in der die Lösung geschrieben steht, möge lieber freigestellt sein. Es ist bestimmt aufschlußreich, wie umständlich / listig die eine oder andere Aufgabe in den verschiedenen Sprachen aussieht.

Ich würde mich sehr freuen wenn das nächste Problem in dieser Form gestellt wäre!

Wie denkt ihr darüber?

Dicki Kasper

-- Termine -- Termine -- Termine --

Nächster Redaktionsschluß28. Dezember 1986
Mitgliedsbeitrag 1987 (letzter Termin) 15. Januar 1987
Jahreshauptversammlung 198713. - 15. März 1987
Norddeutsches Regionaltreffen25. + 26. April 1987

-- Messen '86 --

Hobby-tronikDortmund 18. - 22. Februar 1987
CeBITHannover 4. - 11. März 1987
Internationale Computer-Ausstellung ..Köln 11. - 14. Juni 1987
HUBBY ELEKTRONIK 87Stuttgart 5. - 8. November 1987
PRUDUTRONICAMünchen 10. - 14. November 1987

BASIC-Rätsel

Der Kniff mit der Logik

Wer möglichst kurze und möglichst schnelle Programme schreiben möchte, kommt um logische Verknüpfungen nicht herum. Vor allem, wenn es darum geht, mehrere Wenn-Dann-Bedingungen (IF - THEN) zusammenzufassen. sind entsprechende Verknüpfungen mit AND und OR ratsam. Dabei muß man allerdings darauf achten, wie unterschiedlich sich diese logischen Operanden verhalten. Denn bei AND wird die Anweisung hinter THEN nur dann ausgeführt, wenn alle mit AND verbundenen Bedingungen hinter IF erfüllt sind. Anders bei OR: Es reicht, wenn eine der Bedingungen hinter IF erfüllt ist, und schon wird die Anweisung hinter THEN ausgeführt. Im Beispiel geht es darum, ob ein Ergebnis ausgedruckt werden soll. Der Computer fragt endlos, egal, wie die Antwort lautet. Warum?

```
500 REM DRUCKER-
    ABFRAGE
510 PRINT "AUSGABE AUF
    DRUCKER (Y/N)?"
520 GET TASTES: IF
    TASTES="Y" THEN 520
530 IF TASTES="Y"
    OR TASTES="N"
    THEN 510
540 IF TASTES="Y" THEN
    GOSUB 1000
550 END
1000 PRINT "DRUCK-
    ROUTINE"
```

Auflösung auf S. 111

Computergeschichte(n)

Am Anfang war ein Ungetüm

Die Zuhörer, die sich in einem kleinen Hörsaal der Technischen Universität Berlin 1938 versammelt hatten, glaubten ihren Ohren nicht zu trauen. Da trugen zwei ältere Semester, Konrad Zuse und sein Freund Helmuth Schreyer, nach der Vorführung von Versuchsschaltungen den Gedanken einer elektronischen Rechenmaschine vor. Was die gesetzten Herren Professoren fast noch mehr verwirrte, war die Unbekümmertheit, mit der die zwei ihre Wünsche an Bauteile aufzählten. Mit Kopfschütteln wurde der Bedarf an zweitausend Röhren und einigen tausend Glühlampen quittiert. »Reine Phantasterei – selbst die modernsten elektronischen

Sendeanlagen haben nicht mehr als einige hundert Röhren. Und dann allein der Stromverbrauch von so einem Ungetüm!«

Aber der eigenwillige Erfinder, der seine ersten Versuchsmodelle im ausgeräumten elterlichen Wohnzimmer baute, gab nicht auf. 1940 war Z 2 fertig: ein elektronischer Rechner mit einer Speicherkapazität von 64 Wörtern, programmgesteuert über eine Lochstreifeneingabe.

Das war so revolutionär, daß Zuse sogleich einen Auftrag bekam. Für die Deutsche Versuchsanstalt für Luftfahrt baute er seinen dritten Rechner, den Z 3, der heute als erster echter Computer im Deutschen Museum steht. Nur die Luftfahrtstechniker mit ihren aufwendigen und komplizierten Berechnungen hatten das Zukunftsweisende an Zuses Computern erkannt. Hätte er damals mehr Unterstützung erhalten, würden wir heute vielleicht den Computer nicht als amerikanische Erfindung, sondern als deutsche empfinden.



Der deutsche Computerpionier Konrad Zuse (hinten sein Z3).

HEFT
26
November
1986

86

Ehret die alten Römer!

Keiner weiß heute mehr, wozu die römischen Zahlen gut sind.
Ich auch nicht.

Vielleicht, um den Bildungsgrad von Touristen zu prüfen, die vor ehrwürdigen Bauwerken stehen und angesichts römischer Inschriften über deren Entstehungszeit nachgrübeln...

0123456789 - wer kennt die arabischen Ziffern nicht?

4711 - wer nicht die aus ihnen gebildeten Zahlen?

Aber wer kennt noch: MMMDCCXI ?? (Unter uns: Es ist dasselbe!)

Willkommen wäre also ein Programm, das aus römischen Zahlen arabische zaubert.

Aber das wäre zu einfach! Außerdem wäre das so, als schenkte man uns ein Wörterbuch "Chinesisch/Deutsch", wenn wir nach China reisen - es würde uns nichts helfen, solange wir nicht Chinesisch können! Wir brauchen zuerst ein Wörterbuch "Deutsch/Chinesisch"!

Also fangen wir mit der Übersetzung der arabischen Zahlen in die römischen an. Wer Lust und Mut hat, kann danach das "Anti-Programm" selbst bauen: zur Übersetzung vom Römischen ins Arabische! (Was natürlich viel leichter ist...)

Tschüßchen! * KaJot *

Schon wieder ein "Zehnzeiler"!...

```
1 CLS:PRINT"  Umwandlung von arabischen in roemische Zahlen
=====
"
"      <C> K.J.Muehlenbein, Weinheim 1986
2 DEFINTE,H,N,T,Z:DEFSTRC,D,I,L,M,R,X,V:I="I":V="V":X="X":L=
"L":C="C":D="D":M="M":PRINT"Gib eine ganze Zahl <10000 ein!
":INPUTZ0:T=INT(Z0/1000):IFT>0:FORN=1TOT:R=R+M:NEXT
3 Z1=Z0-1000*T:H=INT(Z1/100):IFH>0:IFH>460SUB7ELSEFORN=1TOH:
R=R+C:NEXT
4 Z2=Z1-100*H:Z=INT(Z2/10):IFZ>0:IFZ>360SUB8ELSEFORN=1TOZ:R=
R+X:NEXT
5 E=Z2-10*Z:IFE>0:IFE>360SUB9ELSEFORN=1TOE:R=R+I:NEXT
6 GOTD10
7 IFH=5R=R+D:RETURN:ELSEIFH<9:R=R+D:FORN=1TOH-5:R=R+C:NEXT:R
ETURN:ELSER=R+C+M:RETURN
8 IFZ=4R=R+X+L:RETURN:ELSER=R+L:IFZ>5FORN=1TOZ-5:R=R+X:NEXT:
RETURNELSERRETURN
9 IFE=4R=R+I+V:RETURN:ELSEIFE<9R=R+V:IFE=5RETURNELSEFORN=1TO
E-5:R=R+I:NEXT:RETURN:ELSER=R+I+X:RETURN
10 PRINTR:PRINT"ALLES KLAR? * ALLES KLAR!":END
```



WIR HABEN EIN SEHR GUTES BETRIEBSKLIMA



Nun müssen wir uns nur noch selbst verkleinern!

Logische Operationen in BASIC 2

Liebe Clubfreunde,

leider mangelt es mir zur Zeit an der nötigen Muße um die im ersten Teil dieses Beitrages angekündigte Fortsetzung zu verfassen. Aber aufgeschoben ist ja nicht aufgehoben, wie schon das alte Sprichwort sagt und so verspreche ich euch die Fortsetzung für eines der nächsten Infos.

Aber wenigstens die sechs fleißigen und schnellen Einsender von Lösungen zu meinem kleinen Preisrätsel sollen nicht länger auf die Folter gespannt werden und endlich erfahren, wer denn nun die Flasche mit dem garantiert glycolfreien Wein gewonnen hat.

Eingegangen sind insgesamt sechs Lösungen. Teilgenommen haben: Klaus Hermann, Eckehard Kuhn, K.J. Mühlenbein, Kurt Müller, Bernd Retzlaff und Arnulf Sopp. Alle eingesandten Lösungen waren richtig.

Die vollständigsten (aber das war ja nicht gefordert) Lösungen stammen von Kajott und Bernd Retzlaff. Beide verschönten ihre richtigen Lösungen noch mit einem entsprechendem Demoprogramm, wobei Bernd immerhin einen drei Seiten langen Brief zusammenbrachte, während Kajott das Ganze auf eine normale Postkarte quetschte (allein das ist schon anerkennenswert!).

Hier die Lösungszeilen:

C=(A OR B) AND (NOT(A AND B))

Kajott

IF (A OR B) AND NOT (A AND B) THEN C=1 ELSE C=0

Bernd

Wie ihr seht, kamen bei gleichem Lösungsweg zwei unterschiedliche BASIC-Zeilen heraus, die jedoch beide ihre Pflicht erfüllen!

Auch Kurt Müller packte seine Lösungszeile in ein Programm und garnierte das Ganze noch zusätzlich mit einer Herleitung der Lösung mittels der Booleschen Algebra und einem Karnoagh-Diagramm (leider habe ich heute keine Zeit euch in die Feinheiten und Geheimnisse dieses Algebraablegers einzuführen, aber vielleicht läßt sich Kurt ja dazu breitschlagen!). Seine Lösungszeile lautet: C=(NOT(A) AND B) OR (A AND NOT (B)).

Zu genau der gleichen Lösung kam Eckehard Kuhn, der allerdings außer der Lösung (und natürlich dem Absender und Empfänger) keine weiteren Worte auf der von ihm benutzten Postkarte verlor (was ja auch nicht gefordert war!).

Gleich zwei Lösungen sandte Klaus Hermann ein und beide funktionierten. Seine Lösungszeilen lauten:

1. C=(A AND NOT B) OR (NOT A AND B)

2. C=NOT(NOT A OR B) OR NOT(A OR NOT B)

Was wieder mal beweist, nicht nur nach Rom führen viele Wege!

Die wohl kürzeste Lösung stammt von unserem Assemblerfreak Arnulf Sopp. Er verwandte neben den logischen Operationen auch eine mathematische und so lautet seine BASIC-Zeile:

C=(A OR B) - (A AND B)

Ich wollte zuerst auch nicht glauben, daß das funktionieren kann. Aber nachdem ich die Zeile flux in das Programm von Kajott eingesetzt hatte, stellte sich die Funktionsfähigkeit sofort heraus!

Sechs Einsendungen und sechs richtige Lösungen (es gibt übrigens noch mehr, so erfüllt die Zeile "C=(A+B) AND 1" auch ihren Zweck) machen mir natürlich die Entscheidung schwer, wer nun den edlen Tropfen bekommen soll. Noch dazu, da alle Lösungen fast gleichzeitig bei mir eintrafen! Aus diesem Grunde habe ich mich zu der, so hoffe ich gerechtesten Entscheidung durchgerungen, die als Preis ausgesetzte Flasche selbst zu leeren! Prost!!!

Damit sich die Teilnehmer aber nicht umsonst so viel Mühe gegeben haben, werde ich zusätzlich jedem der Sechs eine Flasche zukommen lassen und hoffe, daß sie sie auf das Wohl des CLUB 80 trinken werden. Also nochmal Prost!!!

Damit auch die restlichen Leser dieser kleinen Artikelserie, die sich vielleicht schon auf den zweiten Teil gefreut haben, auch nicht zu kurz kommen, folgt hier nun ein Artikel aus der letzten "mc", der sich mit einem ähnlichen Problem beschäftigt. Viel Spaß bei der Lektüre wünscht euch

Karntaut Obermann

Fast hätte ich was vergessen! Leider ist mir im ersten Teil ein kleiner Tipfehler unterlaufen, den ich trotz mehrfacher Kontrolle übersehen habe. Die Wahrheitstafel der AND-Verknüpfung enthält in Zeile vier einen Fehler! Die korrigierte Version sieht so aus:

A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

HEFT
16
November
1986

11

Meinung zur Meinung von Dieter Kasper Seite 5:

Die Rahmenbindung meines Programm-Wettbewerbes auf max. 33 Zeilen geschah aus der Sicht des Auswertenden. In der Hoffnung auf viele Teilnehmer wollte ich vermeiden, einen Stapel ellenlanger Listings auf sinnvolles und ideenreiches Programmieren zu durchsuchen und daraus eine Rangfolge zu bestimmen. BASIC war vorgegeben; was nützt mir das schönste Programm in C, Cobol oder Forth? Ich bin dieser Sprachen nicht mächtig (solls ja auch noch geben). Vielleicht lag es an der (Un-)Lust der Mitglieder? Die Preise waren doch wohl ok! Ein bisschen Rahmenbindung muß bei einem Wettbewerb wohl sein. Sei's drum, es hat trotzdem Spass gemacht. (BitPit)

Logiksimulation mit Basic

Bei jeder Umsetzung logischer Verknüpfungen in eine digitale Schaltung stellt sich die Frage, ob das, was auf dem Papier steht, noch Ähnlichkeit mit der Zielsetzung hat. Wie man ohne großes Softwarepaket das Erstellen einer Wahrheitstafel einem beliebigen Basic-Rechner überträgt, zeigt der folgende Beitrag.

Das Problem ist jedem, der sich mit digitaler Hardware einläßt, sicher vertraut: Man hat sich aus den Karnaugh-Diagrammen oder den logischen Gleichungen eine Schaltung aufgezeichnet, eventuell noch optimiert und Laufzeiten angepaßt, so daß die Gatteranordnung nicht mehr viel mit dem ursprünglichen Entwurf gemeinsam hat. Auf den ersten und oft auch auf den zweiten, etwas gründlicheren Blick läßt sich nicht mehr sagen, ob sich bei all den (natürlich sorgfältig ausgeführten) Veränderungen nicht doch ein Fehler eingeschlichen hat. Besonders undurchsichtig wird das Ganze, wenn man zur Minimierung der Bausteinzahl noch sogenannte „Punktschiebungen“ durchgeführt hat, um alles einheitlich auf NANDs und NORs zu reduzieren – spätestens jetzt ist jede Übersicht verloren. Die Versuchung ist groß, die eigene Fehlerquote bei 0% anzusiedeln und sich an den Aufbau zu wagen. Ist etwas mehr Mut vorhanden und hat man die Tabelle doch aufgestellt, so kommt oft bei mehrmaligem Überprüfen jedesmal eine andere Tabelle heraus, denn auch das Erstellen von Wahrheitstabellen ist eine fehlerträchtige

ge Beschäftigung – und damit eigentlich eine ideale Aufgabe für einen Computer.

Die Basic-Befehle

Schaut man sich den Befehlssatz der handelsüblichen Basic-Interpreter an, so fallen unter der Rubrik „Operatoren“ die Befehle NOT, AND, OR und EXOR auf. Sie werden meist zur Verknüpfung mehrerer Bedingungen in IF-Ausdrücken verwendet, eignen sich jedoch auch vorzüglich zum logischen Rechnen. Dazu muß man nur wissen, daß die logische 1 als -1 dargestellt wird; die Null bleibt, wie sie ist. Eine Wertetabelle stellt die Ausgabewerte für alle möglichen Kombinationen der Eingangssignale dar. Das Durchspielen aller Möglichkeiten läßt sich in jeder höheren Programmiersprache durch das Verschachteln von Schleifen erzielen. Und da Basic beliebige Schrittweiten – also auch negative – zuläßt, ist das Problem eigentlich schon gelöst. Bild 1 zeigt, wie man sich die Wertetabelle eines NAND-Gatters (Bild 2) ausdrucken lassen kann. Die beiden FOR-Schleifen bewirken, daß die Eingänge A

Bild 2. Die Wahrheitstabelle des NAND-Gatters

A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0

und B alle möglichen Kombinationen aus Null und Eins durchlaufen. In der inneren Schleife erfolgt die Verknüpfung und das Ausdrucken der Tabelle. Die Funktion ABS verhindert, daß vor eine logische Eins ein Minus geschrieben wird, denn wie bereits erwähnt rechnet Basic intern mit diesem negativen Wert. Eine Sonderstellung nimmt unter Umständen die EXOR-Funktion ein. Ist sie ordnungsgemäß implementiert (wie z.B. bei MBasic), so ergibt sich die Wahrheitstabelle von Bild 4. Es kann aber auch anders kommen: So ist etwa beim Basic des AIM-65 diese Funktion nicht vorgesehen. Trotzdem meldet der Rechner beim vorliegenden Programm (Bild 3) keinen Fehler und gibt als Ergebnis für Z die Werte von B aus! Die Erklärung liegt in der Tatsache, daß bei diesem Basic Leerzeichen als Trenner nicht erforderlich sind. Aus $Z = A \text{ XOR } B$ wird so für den Rechner $Z = A \text{XOR } B$, und da die Variable AX nicht initialisiert wurde und daher auf Null gesetzt wird, ergibt sich konsequenterweise für Z der Wert von B. Es ist also nicht verkehrt, sich von der ordnungsgemäßen Funktion des XOR-Befehls zu überzeugen und sich gegebenenfalls mit der Umschreibung $Z = (\text{NOT}(A) \text{ AND } B) \text{ OR } (A \text{ AND } \text{NOT}(B))$ zu behelfen.

Ein praktisches Beispiel

Bei den bisher aufgeführten Beispielen kommen natürlich die Fähigkeiten eines Basic-Rechners nicht sonderlich zum Tragen. Daher soll als nächstes ein Beispiel aus der Praxis herangezogen wer-

Bild 4. So sieht die Tabelle bei richtig arbeitendem XOR-Befehl aus

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	0

den. Um aus dem Adreßraum von einem KByte, der von einem Chip-Select zur Verfügung gestellt wird, die ersten 64 Byte für einen zusätzlichen Peripheriebaustein auszublenken, entstand die Schaltung nach Bild 5. Sie ist umgeformt in Bausteine des Typs NAND und NOR, um die Anzahl der Chips zu reduzieren. Im Programm (Bild 6) sieht man, daß die äußerste Schleife die höchstwertigste, die innerste die niederwertigste Leitung enthält. Zusammen mit der Reihenfolge der PRINT-Befehle in der innersten Schleife ergibt sich das gewohnte Bild der Wahrheitstabelle (Bild 7). Um die Gleichungen überschaubar zu halten empfiehlt es sich, Zwischenvariable einzuführen und in der Reihe des Signalflusses hinzuschreiben. Fest auf Null oder Eins gelegte Eingänge kann man entweder in Gleichungen mit aufnehmen, oder man reduziert den entsprechenden Term vorher. Beides findet man in diesem Beispiel. Bild 7 zeigt, daß die Schaltung trotz Transformation noch so arbeitet, wie es

geplant war. Der erste 64-Byte-Raum wird ausgeblendet, C1 wird aktiv. Alle folgenden Räume sind für C2 reserviert, solange CS, das den 1 KByte-Raum selektiert, aktiv ist. Wenn CS inaktiv wird, bleiben auch C1 und C2 gesperrt. Auch dieses Beispiel ist natürlich nicht außergewöhnlich kompliziert, doch es bestehen schon einige Fehlermöglichkeiten.

Rückkopplung

Bislang wurden nur Schaltungen betrachtet, die Signale streng geradeaus verknüpfen. Ganz neue Perspektiven eröffnen sich, wenn man asynchrone Signale rückkoppelt. Obwohl die Mehrzahl solcher Fälle vermieden werden kann (und sollte!), läßt es sich nicht immer umgehen. Solche Verknüpfungen verlangen Aufmerksamkeit, denn sie können mehrere unangenehme Eigenschaften besitzen, wie die folgenden Beispiele zeigen. Interessant ist jedoch, daß sich die damit verbundenen Probleme mit dem bisher beschriebenen Simulationsverfahren ebenfalls handhaben und darstellen lassen. Bild 8 zeigt eine rückgekoppelte Schaltung, Bild 9 das Simulations-Programm und Bild 10 die Wertetabelle. Solche Verknüpfungen lassen sich analysieren, indem man die Rückkopplung auf-

schneidet und dort, wo sie eingespeist wird, einen eigenen Signalnamen vergibt. Dann läßt man die Simulation in der oben beschriebenen Weise ablaufen, jedoch mit vier anstatt mit drei Eingangssignalen. Zur Betrachtung des Verhaltens der Schaltung sind jedoch nur die tatsächlichen Eingänge A, B und C interessant. Sie bilden quasi den Eingangszustand, dessen verschiedene Werte durch die waagerechten Striche getrennt sind.

Wie wird Bild 10 nun ausgewertet? Die Anschlüsse D und G sind natürlich nicht unabhängig, sondern über eine zeitliche Verzögerung verbunden. Besitzen sie in einer Zeile denselben Wert, so ist der Zustand stabil und der Wert des Ausgangs G eindeutig. Dies ist zum Beispiel für den Eingangszustand $ABC = 001$ für beide Wertepaare von D und G gegeben. Allerdings sieht man hier schon die erste Besonderheit, die bei rückgekoppelten Schaltungen auftritt: War der Rückkoppleingang D vorher auf Eins, so geht auch G auf Eins. War er jedoch auf Null, so nimmt auch G diesen Wert an. Beim Übergang nach $ABC = 001$ ergibt sich also der Wert von G aus dem vorherigen Wert von G, das Ergebnis ist also von der Vorgeschichte abhängig. Gleiches gilt auch für den Zustand $ABC = 011$. Alle übrigen Zustände zeigen ein Ver-

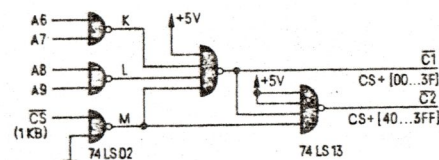


Bild 5. Ein Beispiel aus der Praxis

```

210 REM
220 LPRINT
230 LPRINT "A B Z"
240 LPRINT "-----"
250 : FOR CS=0 TO -1 STEP -1
260 :   FOR A9=0 TO -1 STEP -1
270 :     FOR A8=0 TO -1 STEP -1
280 :       FOR A7=0 TO -1 STEP -1
290 :         FOR A6=0 TO -1 STEP -1
300 :           K = NOT (A6 OR A7)
310 :           L = NOT (A8 OR A9)
320 :           M = NOT (CS OR 0)
330 :           C1 = NOT (K AND L AND M)
340 :           C2 = NOT (C1 AND M AND -1 AND -1)
350 :           LPRINT ABS(CS); ABS(A9); ABS(A8); ABS(A7);
360 :           LPRINT ABS(A6); ABS(C1); ABS(C2)
370 :         NEXT A6
380 :       NEXT A7
390 :     NEXT A8
400 :   NEXT A9
410 : NEXT CS
420 LPRINT
430 LPRINT "Wahrheitstafel fuer Beispiel 3"
440 LPRINT
450 END

```

Bild 6. Das Programm zur Simulation der Schaltung aus Bild 5

CS	A9	A8	A7	A6	C1	C2
0	0	0	0	0	0	1
0	0	0	0	0	1	0
0	0	0	0	1	0	0
0	0	0	0	1	1	0
0	0	0	1	0	0	1
0	0	0	1	0	1	0
0	0	0	1	1	0	0
0	0	0	1	1	1	0
0	0	1	0	0	0	1
0	0	1	0	0	1	0
0	0	1	0	1	0	0
0	0	1	0	1	1	0
0	0	1	1	0	0	1
0	0	1	1	0	1	0
0	0	1	1	1	0	0
0	0	1	1	1	1	0
1	0	0	0	0	0	1
1	0	0	0	0	1	0
1	0	0	0	1	0	0
1	0	0	0	1	1	0
1	0	0	1	0	0	1
1	0	0	1	0	1	0
1	0	0	1	1	0	0
1	0	0	1	1	1	0
1	0	1	0	0	0	1
1	0	1	0	0	1	0
1	0	1	0	1	0	0
1	0	1	0	1	1	0
1	0	1	1	0	0	1
1	0	1	1	0	1	0
1	0	1	1	1	0	0
1	0	1	1	1	1	0
1	1	0	0	0	0	1
1	1	0	0	0	1	0
1	1	0	0	1	0	0
1	1	0	0	1	1	0
1	1	0	1	0	0	1
1	1	0	1	0	1	0
1	1	0	1	1	0	0
1	1	0	1	1	1	0

Bild 7. Die Tabelle zeigt das ordnungsgemäße Arbeiten der Schaltung aus Bild 5

```

210 REM
220 LPRINT
230 LPRINT "A B Z"
240 LPRINT "-----"
250 : FOR A=0 TO -1 STEP -1
260 :   FOR B=0 TO -1 STEP -1
270 :     Z = NOT(A AND B)
280 :     LPRINT ABS(A); ABS(B); ABS(Z)
290 :   NEXT B
300 : NEXT A
310 LPRINT
320 LPRINT "Wahrheitstafel von Beispiel 1"
330 LPRINT
340 END

```

Bild 1. Dieses Programm druckt die Wahrheitstabelle eines NAND-Gatters

```

210 LPRINT
220 LPRINT "A B Z"
230 LPRINT "-----"
240 : FOR A=0 TO -1 STEP -1
250 :   FOR B=0 TO -1 STEP -1
260 :     Z = A XOR B
270 :     LPRINT ABS(A); ABS(B); ABS(Z)
280 :   NEXT B
290 : NEXT A
300 LPRINT
310 LPRINT "Wahrheitstafel fuer Beispiel 2"
320 LPRINT
330 END

```

Bild 3. Von der richtigen Wirkung des XOR-Befehls sollte man sich mit diesem Programm überzeugen

Bild 8. Rückgekoppelte Logikschaltungen wie diese sind auf den ersten Blick immer ein wenig undurchsichtig

```

220 LPRINT
230 LPRINT " A B C D E F G "
240 LPRINT "-----"
250 : FOR A=0 TO -1 STEP -1
260 :   FOR B=0 TO -1 STEP -1
270 :     FOR C=0 TO -1 STEP -1
280 :       FOR D=0 TO -1 STEP -1
290 :         E = (A AND B)
300 :         F = (C AND D)
310 :         G = (A OR E OR F)
320 :         LPRINT ABS(A); ABS(B); ABS(C); ABS(D);
330 :         LPRINT ABS(E); ABS(F); ABS(G);
340 :         IF G <> D THEN LPRINT " " : GOTO 360
350 :         IF G = D THEN LPRINT
360 :       NEXT D
370 :     NEXT C
380 :   NEXT B
390 : NEXT A
400 : NEXT A
410 LPRINT
420 LPRINT "Wahrheitstafel fuer Beispiel 4"
430 LPRINT
440 END

```

Bild 9. Mit dieser Simulation wird die Schaltung von Bild 8 durchschaubar

A	B	C	D	E	F	G
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	0	1	1
0	1	0	0	0	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	1	1	0	1	1
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	0	1
1	0	1	1	0	1	1
1	1	0	0	1	0	1
1	1	0	1	1	0	1
1	1	1	0	1	0	1
1	1	1	1	1	1	1

Bild 10. Nach kurzem Einschwingen verhält sich die Schaltung von Bild 8 stabil

halten, das nach kurzem Einschwingen in einem eindeutigen Wert für G mündet. Dies sei am Beispiel ABC = 000 gezeigt: Wenn G vor dem Übergang auf diesen Eingangswert gleich Null war, dann gilt dies auch für D. In diesem Fall folgt sofort ein stabiler Zustand für den Ausgang G. War G jedoch vor dem Übergang auf Eins, dann wird G wie vorher auch Null, jedoch wird diese Null als Eingangswert D eingegeben, und nach einem zweiten Durchlauf befindet sich die Schaltung im Ruhezustand. Genauso verhalten sich auch alle anderen Zustände, wobei im Zustand ABC = 100 die Schaltung für G = 1 zur Ruhe kommt. Das Kennzeichnen der kurzzeitig instabilen Zustände wird in den Programmzeilen 340 und 350 vorgenommen.

Schwingungen

Noch kritischer verhält sich die Schaltung in Bild 11, 12 und 13: Hier schwingt G zwar für alle Zustände nach kurzer Zeit ein, und eine Abhängigkeit vom vorangegangenen Wert ergibt sich in keinem Zustand, dafür zeigt aber ABC = 111 ein außerordentlich unschönes Verhalten: Die Schaltung schwingt! Man sieht das sofort daran, daß D und G immer verschiedene Werte haben. Daraus ergibt sich, daß G eine Eins rückkoppelt, wenn D gerade Null war und umgekehrt. Dieser Zustand ist demzufolge unbrauchbar, wenn nicht gerade ein Oszillator aufgebaut werden sollte. Im allgemeinen empfiehlt es sich, die Schaltung neu zu überarbeiten.

Die aufgeführten Beispiele decken in etwa den Problembereich ab, der beim Entwurf von asynchronen Digitalschaltungen im Bereich der Logik auftreten kann.

Selbstverständlich müssen bei einem gründlichen Design noch weitere Anforderungen überprüft werden. Dazu zählen unter anderem Laufzeitprobleme, die nicht mit solch einfachen Hilfsmitteln erfaßt werden können. Nicht umsonst ist zum Thema Logik-Simulation eine schnell wachsende Industrie vornehmlich auf dem CAE/CAD-Sektor entstanden. In vielen Fällen wird die vorgestellte Lösung jedoch ausreichen; im Einzelfall können die Spezialprobleme immer noch einer separaten Betrachtung unterworfen werden.

Bild 11. Nochmal Rückgekoppelt: Trotz der wenigen Elemente neigt diese Schaltung zum Schwingen

```

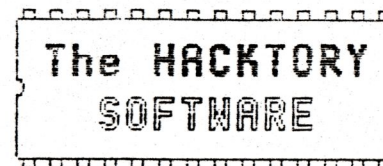
220 LPRINT
230 LPRINT " A B C D E F G "
240 LPRINT "-----"
250 : FOR A=0 TO -1 STEP -1
260 :   FOR B=0 TO -1 STEP -1
270 :     FOR C=0 TO -1 STEP -1
280 :       FOR D=0 TO -1 STEP -1
290 :         E = (A AND B)
300 :         F = NOT (C AND D)
310 :         G = (E AND F)
320 :         LPRINT ABS(A); ABS(B); ABS(C); ABS(D);
330 :         LPRINT ABS(E); ABS(F); ABS(G);
340 :         IF G <> D THEN LPRINT " " : GOTO 360
350 :         IF G = D THEN LPRINT
360 :       NEXT D
370 :     NEXT C
380 :   NEXT B
390 : NEXT A
400 : NEXT A
410 LPRINT
420 LPRINT "Wahrheitstafel fuer Beispiel 5"
430 LPRINT
440 END

```

Bild 12. Das Simulationsprogramm zu Bild 11

A	B	C	D	E	F	G
0	0	0	0	0	1	0
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	0	0	0
0	1	0	0	0	1	0
0	1	0	1	0	1	0
0	1	1	0	0	1	0
0	1	1	1	0	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	0
1	0	1	0	0	1	0
1	0	1	1	0	0	0
1	1	0	0	0	1	1
1	1	0	1	1	1	1
1	1	1	0	1	1	1
1	1	1	1	1	0	0

Bild 13. Die Wahrheitstafel der Schaltung von Bild 11



Arnulf Sopp, Wakenitzstr. 8, D-2400 Lübeck 1

Herrn
Hartmut Obermann
Schwalbacher Str. 6

6209 Heidenrod-Kemel

A. SOPP

Wakenitzstr. 8

2400 Lübeck 1

Telefon:

0451-79 19 26

Ihr Schreiben: 31. 10. 86
Ihr Zeichen:
Mein Schreiben:
Mein Zeichen:
Datum: 3. 11. 86

Auch Prost!

Lieber Hartmut,

man soll nicht glauben, daß so ein süffiger Tropfen noch ohne Glykol hergestellt werden kann. Wer packt das heute noch? Ganz herzlichen Dank dafür! Keine Frage, daß ich ihn auf das Wohl des Clubs und jeden dritten Schluck auf Dein spezielles einnehme.

Über den Auflösungsartikel für das nächste Info mußten wir reichlich lachen, mein Bauch und ich. Insb. daß Du Dir in Deiner Ratlosigkeit den Preis kurzerhand selber zuerkennst, zeugt sowohl von Humor als auch von einer vernaschten Zunge. Bei dem Stoff kein Wunder.

Bei den eingesandten Lösungen frage ich mich, wie ein Mensch mit einem Kopf durchschnittlicher Größe auf sowas kommen kann. Ohne Probelauf mit anschließend angeschlossenem Debugger zum Belauern der Registerinhalte oder was fühle ich mich sogar außerstande, sie zu verstehen. Ich begreife nur, daß Bernd und - Mh ... - Hartmut falsche Lösungen anbieten. So trullt dem Spender das Lala auch steigt, aber eine Lösung, die nur die Ergebnisse 0 und 1 zuläßt, kann nicht stimmen. Bei Bernd ist diese Alternative links und rechts vom ELSE eindeutig. Bei Dir nicht, aber:

Jede Zahl AND 1 kann je nach Bit 0 nur 0 oder 1 betragen, denn der Operand 1 hat nur das Bit 0 auf 1, der Rest ist 0. Nach der Wahrheitstabelle kann demzufolge kein anderes Ergebnis herauskommen, weil bei diesem Operanden alle anderen Partnerbits eine 0 antreffen. 0 AND ir-

HEFT
16
November
1986

gendwas = 0, und das siebenmal. Nur beim allererstenmal (Bit 0) hat die 1 noch Chancen. Und 0000.0000.0000.0001b (BASIC-Integerzahl) ist nun mal = 1.

Beim (simulierten; oder sagt man emulierten?) XOR muß aber jeder Wert zwischen 0 und 255 bzw. 65535 möglich sein. Beispiel:

```
0000.1111.0000.1111b
XOR 1100.1100.1100.1100b
= 1100.0011.1100.0011b = C3C3h = 50115d
```

aber (A sei meinetwegen = 5555h, B = 1C72h):

```
0101.0101.0101.0101b (A)
+ 0001.1100.0111.0010b (B)
= 0111.0001.1100.0111b (A+B)
AND 0000.0000.0000.0001b (1)
= 0000.0000.0000.0001b = 1d/h/o/dingsbums
```

Nimm es zwar ernst, aber nicht tragisch; es geht mir nur darum, mit al-
ler Gewalt recht zu behalten. Aber Jux beiseite, falls ich nicht irre
(Kajott wäre vielleicht der geeignete Prüfstein meiner Behauptung),
solltest Du das unbedingt rechtzeitig vor der Veröffentlichung korri-
gieren, denn unter den Clubmitgliedern sind außer den Teilnehmern viel-
leicht noch dreieinhalb, die das merken. Du hast einen Ruf (nach Har-
vard?) zu verlieren! Bernds Buddel kannst Du ja als Anerkennungspreis
für den fleißigen Teilnehmer deklarieren, der nur leider auf der fal-
schen Fährte landete. Wenn in dem von Dir angekündigten mc-Artikel auch
sowas steht, gäbe das Stoff für eine schöne Glosse über Redaktionen, die
allenfalls Kommafehler bemerken.

Das (nach meiner Theorie) Mißverständnis ist möglicherweise darin
begründet, daß solche Wahrheitstabellen nur ein Bit beschreiben. Dabei
muß man aber im Auge haben, daß dieses eine Bit nur Beispiel für alle
Übrigen ist. Um nun nicht eine Tabelle mit 256² oder gar noch mehr
verschiedenen Operanden-Paarungen auf ein lausiges Stück DIN A4 gnubbeln
zu müssen, unterwirft man sich dieser Beschränkung und den Leser der
seinigen. Pardon!

Welches Beispielprogramm Kajott abgeliefert hat, weiß ich nicht.
Nach den Resultaten bin ich offen gestanden ein bißchen skeptisch. Mein
Beispielprogramm ist anschließend gelistet. Der ML-Teil (da fühle ich
mich sicherer als in BASIC) ist absolut wasserdicht. Die CALLs nach 0A7F
und 0A9A werden im User's Manual zur Übergabe und -nahme von Parametern
an/von ML-Programmen angegeben. Was da herauskommt, ist ein bombensi-
cheres XOR. Es wird in Zeile 60 angezeigt. Nun kannst Du alle Lösungen
als Zeile 70 mit anschließendem PRINT C eintippen. Dann kannst Du mit
ein paar Testoperanden herumspielen und sehen, ob das XOR-Ergebnis mit
dem Lösungsvorschlag übereinstimmt. Wehe, wenn nicht!

Für das nächste Glas von Deinem köstlichen Dope brauche ich die
nötige Andacht. Deshalb Schlunz jetzt und EOF.

Dein



XORzismus

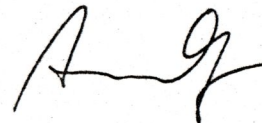
Hallo Hartmut!

Nachträglich fiel mir noch auf, daß das Bit 15 bei Integerzahlen
als Vorzeichen interpretiert wird. Daraus schloß ich messerscharf, daß
der in der ML-Routine höherwertige Operand max. 127 betragen darf, sonst
ist das Ergebnis zwar richtig, aber wegen des Minuszeichens schwierig zu
interpretieren. Also probierte ich das vorsichtshalber mal eben durch
und stellte fest, daß sogar beide Operanden höchstens 127 betragen dür-
fen, sonst wird Überlauf gemeldet. Das ist aber nur eine Besonderheit
des BASIC, mit der diese Testroutine nicht klarkommt. Sonst wäre es we-
sentlich komplizierter geworden. Es ändert nichts an der Tauglichkeit
der Routine, denn bereits 2 ist mehr als 1; es ging ja darum, zu zeigen,
daß die beiden (oder mehr?) falschen Lösungen nur max. 1 hervorbringen.

Das mußte ich noch nachtragen, damit der Test bei der Wahl unge-
eigneter Operanden nicht plötzlich "beweist", daß alle Lösungen falsch
sind.

Ob der Wein damit zu tun hat, daß ich gestern Abend nicht zuende
dachte? Sei's drum, er war die Blamage wert!

Mit gelallten Grüßen, Dein



10 CMD"load,ätsch/cmd"	USR-Routine laden
20 INPUT"A (max. 255):";A'	1. Operand
30 INPUT"B (max. 255):";B'	2. Operand
40 C=A*256+B'	2 X 8 Bit für XOR
50 DEFUSR0=&HF000'	XOR-Operation durchführen
60 PRINTUSR0(C)'	Ergebnis verraten

00001 ;*****			
00002 ;			ATSCH/CMD
00003 ;			Routine zur Entlarvung von
00004 ;			alkoholsüchtigen Wein-Erschleichern
00005 ;			(C) 1986 by Don Promillo
00006 ;*****			
00007 ;			
F000 00008	ORG	0f000h	
F000 CD7FOA	CALL	0a7fh	;HL <- USR-Argument
F003 7C	LD	A,H	;1. Operand
F004 AD	XOR	L	;2. Operand, Erg. -> A
F005 6F	LD	L,A	;HL mit Ergebnis laden
F006 2600	LD	H,00h	;MSB = 0, weil nur 8 Bit
F008 C39A0A	JP	0a9ah	;Argument an BASIC überg.
0000 00015	END		

00000 Fehler

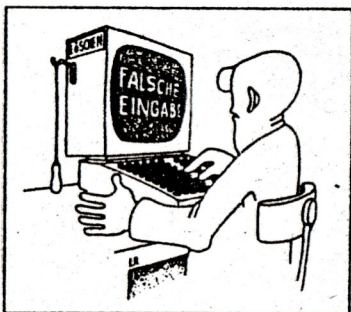
Liebe Freunde,

natürlich hat unser Bitfummelmeister Arnulf recht mit dem, was er über Bernd's und meine Lösung (es ist übrigens eine von mehreren die ich ausprobiert habe!) sagt. Geirrt hat er sich allerdings wenn er glaubt, mir wäre dieser "Fehler" nicht aufgefallen. In dem Programm für welches ich die XOR-Funktion eigentlich brauchte, ist mir die Unbrauchbarkeit dieser Lösung sofort aufgefallen. Trotzdem freue ich mich über die prompte Reaktion und scheue mich nicht, Arnulfs Brief (gekürzt!) und damit meinen Fehler sofort zu veröffentlichen, vor allem da dadurch doch noch eine Fortsetzung von "Logische Operationen in BASIC" erscheint!

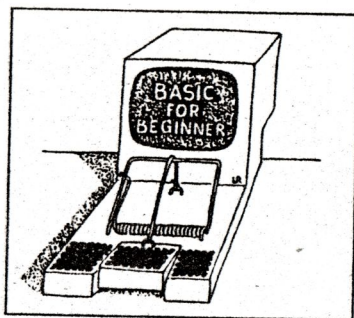
Vielen Dank dafür an Arnulf und alles Gute für alle Clubmitglieder, euer

Karsten Obermann

Weitere Informationen über die
Digitaltechnik in unserer Sonderheft
am Ende des INFO's
Die Redaktion



Frage an den Computer: »Kann ein Computer auch Witze machen?«
Antwort des Computers: »Selbstverständlich. Er kann nur nicht drüber lachen.«



BASIC – Laufende Programme beliebig erweitern

Mit der hier vorgestellten Routine läßt sich ein bereits laufendes BASIC-Programm um eine oder mehrere Programmzeilen erweitern. Ändert man normalerweise ein Programm, dann muß der BASIC-Interpreter angehalten werden. Mit einem RUN-Befehl läßt sich dann das modifizierte Programm wieder starten, allerdings mit einem Schönheitsfehler: Die Variablen des ursprünglichen Programmes sind gelöscht. Ganz anders bei der hier vorgestellten Routine: Sie ist selbst ein Teil des momentan laufenden BASIC-Programmes und kann somit dessen Text erweitern, ohne daß die Variablen in der vom BASIC-Interpreter angefertigten Tabelle überschrieben werden.

ste Zeilennummer größer ist als die letzte vorhandene im laufenden BASIC-Programm. Nur so kann der MERGE-Befehl korrekt arbeiten.

Bei der Programm-Erweiterung entsteht auf der Diskette eine temporäre Datei mit dem Namen „INTER“. In der Zeile 130 fragt deshalb die Routine, ob diese Datei wieder gelöscht werden soll.

Ursprünglich ist das Hilfsprogramm auf einem TRS80-Computer unter Newdos entworfen worden. Es läßt sich aber auch ohne Probleme an fast alle anderen Computer anpassen, deren BASIC-Interpreter über einen MERGE-Befehl verfügt.

Vorteilhaft kann man auch das

```
10 CLS:DIMPZ$(500):CLEAR$000
20 INPUT"Wieviele Programmzeilen sollen angehaengt werden ";N
30 INPUT"Zeilennummer der ersten anzuhaengenden Zeile ";Z
35 INPUT"Abstand zwischen den neuen Programmzeilen ";I
40 FORA=1TON
50 PRINTA"- te Zeile eingeben ( Zeile"Z;"");:INPUT$(A)
60 P$(A)=STR$(Z)+ " "+P$(A)+CHR$(13)
70 Z=Z+I:NEXTA
80 OPEN"O",1,"INTER"
90 FORA=1TON
100 PRINT#1,P$(A);
110 NEXTA:CLOSE
120 MERGE"INTER"
130 INPUT"Inter-File loeschen (J/N) ";D$
140 IFD$="J"THENKILL"INTER"
150 END: REM ***** ODER WEITERES PROGRAMM *****
```

In den Zeilen 20 bis 30 fragt die Routine nach der Anzahl der anzuhängenden Zeilen sowie der Startzeile. Die zuvor eingegebenen Zeilennummern werden dann mit dem Zeileninhalt in Zeichenketten umgewandelt. Anschließend erweitert das Programm jede Zeichenkette um ein Leer- und ein Return-Zeichen und erzeugt somit eine komplette BASIC-Zeile.

Sind alle neuen Zeilen eingegeben, dann werden sie im ASCII-Format auf die Diskette gerettet. Der MERGE-Befehl in der Zeile 120 fügt die zuvor gespeicherten Zeilen dem momentan laufenden BASIC-Programm an. Auf die so angehängten Zeilen kann dann BASIC über eine GOTO- oder GOSUB-Instruktion zugreifen.

Beim Anfügen der Zeilen ist jedoch darauf zu achten, daß die er-

kurze BASIC-Programm als Unterprogramm schreiben, das dann ein laufendes Programm beliebig oft aufruft. So lassen sich zum Beispiel recht leicht bei der Programmentwicklung neue mathematische Funktionen definieren, die dann als gewöhnliche Programmzeilen am Ende des BASIC-Programmes stehen. Oder eine andere interessante Anwendung: Im Speicher steht ein Maschinensprache-Programm. Es läßt sich über PEEK-Instruktionen lesen und anschließend als DATA-Zeile an das Programmende anfügen. Die Umrechnung der hexadezimalen Werte in dezimale entfällt, da diese Arbeit die hier vorgestellte Routine automatisch erledigt.

Ralf Krenzke

HEFT
16
November
1986

18


```

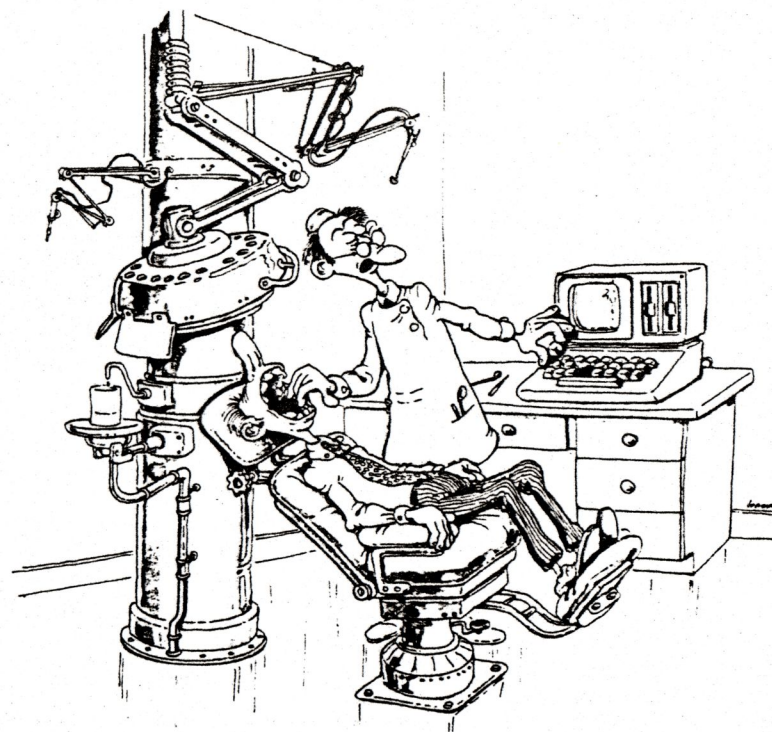
100 ' Spline Interpolation (kubisch)          10.08.1986
110 ' BEDEUTUNG:  Zu gegebenen Punkten...
111 '           >> X: 1, 4, 6, 7          ... N = 4
112 '           >> Y: 0, 1, 6, 3
113 '           ( X-Werte müssen aufsteigend sein !! )
114 '           ...eine möglichst glatte Kurve durch sie zu
115 '           legen. Diese Methode ist wesentlich besser
116 '           als die bekannte Polynominterpolation !
117 '
118 ' Algorithmus natürlich nicht von mir.
119 '
120 '
121 ' Dieter Kasper
122 ' Zeppelinstr. 9
123 ' 8952 Marktoberdorf
124 ' Tel.: 08342 / 1630
125 '
126 '
127 ' N+1: Anzahl der Stützpunkte
128 READ N:N=N-1
129 DIM X(N),Y(N) : 'Stützpunkte
130 DIM D(N),R(N),B(N),C(N),Z(N) : 'Hilfsfelder
131 DIM XX(128),YY(128): Wert des SPLINES an der Stelle (XX,YY)
132 '
133 ' Einlesen der Stützpunkte
134 FOR I=0 TO N: READ X(I) : NEXT
135 FOR I=0 TO N: READ Y(I) : NEXT
136 'FOR I=0 TO N: INPUT " X , Y";X(I),Y(I):NEXT
137 '
138 ' Aufstellen des Tridiagonalen Gleichungssystems
139 ' und Lösen desselben
140 D(0) = X(1)-X(0)
141 FOR I=1 TO N-1: D(I) = X(I+1)-X(I)
142 : C(I) = 2*(D(I-1)+D(I)): NEXT
143 Z(2)=D(1)/C(1)
144 FOR I=3 TO N-1: Z(I)=D(I-1)/(C(I-1)-D(I-2)*Z(I-1)):NEXT
145 FOR I=1 TO N-1: R(I)=(Y(I+1)-Y(I))/D(I)-(Y(I)-Y(I-1))/D(I-1):
146 : R(I)=3*R(I): NEXT
147 B(1)=R(1)/C(1)
148 FOR I=2 TO N-1: B(I)=(R(I)-D(I-1)*B(I-1))/(C(I)-D(I-1)*Z(I)):
149 : NEXT
150 C(N-1)=B(N-1)
151 FOR I=N-2 TO 1 STEP -1: C(I) = (B(I)-Z(I+1)*C(I+1)): NEXT
152 C(0)=0: C(N)=0
153 FOR I=0 TO N-1: B(I)=(Y(I+1)-Y(I))/D(I)
154 : B(I)=B(I)-((2*C(I)+C(I+1))*D(I)/3)
155 : D(I)=(C(I+1)-C(I))/(3*D(I)): NEXT
156 ' An welchen Stellen soll die Splinefunktion ausgewertet
157 ' werden ?
158 W = (X(N)-X(0))/127
159 XX(0)=X(0):FOR I=1 TO 127 : XX(I)=XX(I-1)+W : NEXT
160 '
161 ' Auswertung des Splines ( X-Werte zwischen x(0) und x(n)!)
162 FOR J=0 TO 127: XX=XX(J)
163 FOR I=0 TO N-1
164 IF (X(I+1)-XX)>=0 THEN XX=XX-X(I):
165 : YY(J)=Y(I)+XX*(B(I)+XX*(C(I)+XX*D(I)))
166 : GOTO 520
167 '
168 NEXT I
169 NEXT J
170 ' Zeichnen der Kurve
171 IN=YY(0):SU=IN
172 FOR I=1 TO 127:Y=YY(I): IF Y>SU THEN SU=Y :GOTO 580
173 : IF Y<IN THEN IN=Y
174 : NEXT
175 FOR I=1 TO N:Y=Y(I): IF Y>SU THEN SU=Y :GOTO 610
176 : IF Y<IN THEN IN=Y
177 '

```

```

510 NEXT
520 CLS
530 FOR I=0 TO N: Y=Y(I): GOSUB 700
540 : K = (X(I)-X(0))/W
550 : FOR J=47 TO Z STEP -1: SET (K,J): NEXT J,I
560 FOR I=0 TO 127 : Y=YY(I): GOSUB 700
570 : SET (I,Z): NEXT
580 GOTO 580
590 END
700 Z=INT(47*(SU-Y)/(SU-IN)):RETURN
710 DATA 10
720 DATA 1,2,3,4,5,6,7,8,9,10
730 DATA 0,0,0,0,0,9,9,9,9,9

```



Endlich weiß ich, wozu dies verdammte Ding zu gebrauchen ist.

* Goldener FIBONACCI ! *

Her mit Papier, Bleistift und Lineal! Laßt uns eine Strecke zeichnen von beliebiger Länge. Nennen wir diese Länge "1"; sie wird unser "persönlicher" Maßstab! Markieren wir ihre Enden.

Jetzt verlängern wir diese Strecke um ein Stück s . s soll kürzer sein als 1, und zwar nur so lang, daß es zur Strecke 1 im gleichen Verhältnis steht wie die Strecke 1 zur Gesamtstrecke $1+s$; so daß gilt:

$$s:1 = 1:(1+s) \quad \text{oder kurz} \quad s = \frac{1}{1+s}$$

Umformung ergibt die quadratische Gleichung: $s^2 + s = 1$.
Daraus erhalten wir als (positive) Lösung:

$$s = (\sqrt{5} - 1)/2 = 0,618039...$$

also natürlich eine Irrationalzahl (wegen der Wurzel aus einer Zahl, die kein Integer-Quadrat ist); aber das stört uns nicht, wir brechen sie ab, wo es uns Spaß macht. Z.B. wie hier nach der 6. Dezimalstelle. Oder auch erst nach der 15.!

Wieso 15? Wer wird so lange an der Wurzel herumrechnen?
Na wat'n, wozu ham'wa denn'n Komputa?? sacht da Berlina!

Also: laßt uns ein Programm "entwickeln!" - Aber nicht eins, das eine genauere Wurzel aus 5 zieht - das wäre uninteressant - sondern eins, das sich die Gesetzmäßigkeit zunutze macht, die in unserer Ableitung des Streckenverhältnisses 0,618039...:1 steckt!

Dies ist nämlich nichts anderes als der berühmte
"Goldene Schnitt"

den sich viele Maler, Bildhauer, Architekten und andere als Leitbild (Leitmaß!) genommen haben. Dieser liegt dann vor, wenn sich eine Strecke zu einer größeren genau so verhält wie die größere ihrerseits zur Gesamtlänge, also zur Summe beider Teilstrecken.

Setzen wir dies Spielchen fort!

Wir fügen zur Gesamtlänge $(1+s)$ wieder ein Stückchen hinzu, das zu $(1+s)$ im gleichen Verhältnis steht wie $s:1$. Und siehe da - dann steht auch $(1+s)$ zu dieser neuen Gesamtlänge wieder in dem gleichen Verhältnis! Wie lang ist diese neue Gesamtlänge? Natürlich $(1+s)*(1+s)!$ Die Formel für jede neue Länge - wenn wir so fortfahren - lautet: $s_n = (1+s)^n$.

Aber uns interessieren nicht die Längen, sondern nur deren Verhältnisse untereinander. * Vergewenwärgen wir uns noch einmal das Vorgehen. Zuerst galt doch:

$$s:1 = 1:(s+1), \text{ in Worten:}$$

Der Zähler des neuen Bruches verhält sich zu seinem Nenner wie der Nenner des vorhergehenden Bruches zu dessen Summe Zähler+Nenner. Der alte Nenner wird also neuer Zähler und die Summe (alter Zähler+Nenner) wird neuer Nenner.

Ganz einfach". Was kommt da heraus? (Kürzen wir ab: Zähler=Z, Nenner=N, alt=a, neu=n)

(1) $nZ = aN =$	$\frac{1}{s+1}$
$nN = aZ (s) + aN (1) =$	
(2) $nZ = aN =$	$\frac{s+1}{s+2}$
$nN = aZ (1) + aN (s+1) =$	
(3) $nZ = aN =$	$\frac{s+2}{2s+3}$
$nN = aZ (s+1) + aN (s+2) =$	
(4) $nZ = aN =$	$\frac{2s+3}{3s+5}$
$nN = aZ (s+2) + aN (2s+3) =$	
(5) $nZ = aN =$	$\frac{3s+5}{5s+8}$
$nN = aZ (2s+3) + aN (3s+5) =$	
(6)	$\frac{5s+8}{8s+13}$
(7)	$\frac{8s+13}{13s+21}$
(8)	$\frac{13s+21}{21s+34}$
	u.s.w.

Die Zahlenfolge, die sich hier deutlich abhebt, heißt die
FIBONACCI-Folge

nach dem italienischen Mathematiker Leonardo Pisano Fibonacci (ca. 1180-1240). Sie entsteht, wenn man mit 0 und 1 beginnt und zur letzten immer die vorletzte Zahl addiert (Rekursionsformel):

0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765 10946 17711

Das n -te Glied lautet: $0,4472136[1,618034^n - (-0,618034)^n]$.

Bildet man das Verhältnis zweier Nachbarglieder, so stimmt es um so besser mit dem Goldenen Schnitt überein, je weiter man in der Folge voranschreitet - der Fehler verschwindet aber nie ganz, weil in Zähler und Nenner immer das "Korrekturglied" s fehlt. Es ist dann so, als hätte man in obigen Brüchen $s=0$ oder $s=1$ gesetzt. Daß dem so ist, zeigt das nachstehende Programm recht deutlich. Die Quotientenfolge konvergiert erst im Unendlichen gegen s . [Wegen der Genauigkeit unseres Computers konvergiert sie ab 29. Glied nicht mehr.]

So ist also FIBONACCI das mathematische Pendant zu den zahlreichen Alchimisten des Mittelalters, jenen Chemikern, die sich unzählige Nächte vergeblich abmühten, aus dem "Stein der Weisen" GOLD zu machen: Er hat es fast geschafft! Wie diese war er nahe dran - hätte er nur schon an den "Goldenen Schnitt" gedacht! * Aber das wäre so gewesen, als hätte ein Alchimist Aurum in seine Brüche gegossen und dann gerufen: ἐνσταλά ! Und sowas macht man nicht!

- Also sprach $\alpha\omega\tau\ \mu\beta$

~

Der Mathematiker wird bewert haben, daß die Ableitung nicht ganz zwingend war. Der saubere Beweis geht einen anderen Weg, den ich mir hier aber verkneifen habe, weil wir uns ja nicht in einen Lehrgang über Zahlentheorie befinden. - Hugh!

HEFT
16
November
1986

12


```

10 CLS
  PRINT CHR$(23) "      Die FIBONACCI-Reihe"
  FOR I = 1 TO 300
  NEXT
  PRINT CHR$(28)
20 CLEAR 70
  DEFINT I
  DEFDBL V,N,Z
  DEFSTR F,S
  S = STRING$(63,"-")
  N = 1
  F = ".6180339887"
  PRINT "Wieviele Rekursionsschritte ? ";
  INPUT "(Bitte nur gerade Anzahl eingeben)";IT
  ZZ = 2
  FOR IO = 1 TO IT STEP 20
30 CLS
  PRINT " V      N      Quotient V/N"
  PRINT S
40 PRINT "'Goldener Schnitt' = Grenzwert: "F
  PRINT S
50 IF IO = 1
  THEN PRINT V,N
60 FOR I = 1 TO 20
  ZZ = ZZ + 1
  IF ZZ <= IT
  THEN Z = N
      N = N + V
      ON (2 + (2 * INT (I / 2) = 1)) GOSUB 80,70
      V = Z
      NEXT I
      INPUT "weiter <ENTER> ";E$
      NEXT IO
  END
  ELSE END
70 PRINT N,
  RETURN
80 PRINT N,Z / N
  RETURN

```

[... ein Achtzeiler !]

Lewis ROSENFELDER zeigt in "BASIC FASTER AND BETTER & other mysteries", erschienen 1981 bei IJG Inc. (existiert m.W. nicht mehr), ISBN 0 936200 030, Möglichkeiten im Umgang mit BASIC (z.T. unter Verwendung von "Maschinensprache"), die hier und da auch den geübten und erfahrenen Programmierer überraschen und bereichern. Es ist daher sehr bedauerlich, daß die in unserem INFO begonnene recht gute Übersetzung des - immerhin fast 300 Seiten langen - Buches nicht vollendet werden konnte. (Ehrlich: Ich möchte diese Herkulesarbeit nicht machen!) Lest einmal seine Erläuterungen zum Aufbau von "Logik-Kürzeln", wie ich es einmal nennen möchte. Seit ich das las, verwende ich sie mit Vorliebe.

Das Prinzip besteht darin, algebraische Beziehungen (Gleichungen/Ungleichungen) in logische Wahrheitswerte
 logisch "WAHR" = 1 (bzw. im BASIC: -1)
 logisch "NICHT-WAHR" = 0
 umzusetzen.

Zwischenbemerkung

"Logisch Nicht-Wahr" ist übrigens nicht etwa dasselbe wie "logisch falsch"! "Logisch falsch" ist schlichtweg unlogisch und nichts weiter wie ein Denkfehler; es stellt sich quer, d.h. ein Programm, das eine Unlogik enthält, läuft erst gar nicht. Man muß den Fehler suchen. Wenn ich schreibe "2 * 2 = 3.9", so ist das zwar "falsch", aber nicht "unlogisch": die Logik besteht hier darin, daß eine Antwort jedenfalls möglich ist. Sie lautet (ohne besondere Lautstärke): "Das ist nicht WAHR! Deine 'Gleichung' hat den 'Wahrheitswert Null'!" Das Programm antwortet allerdings kürzer: einfach mit "0". Hingegen wäre es unlogisch zu schreiben: "2 * 2 = VIER" oder "2 * 2 = WEISSNICH", denn numerische Variable können nicht einem String gleichgesetzt werden. Dieses Programm "läuft" nicht; es folgt ==>ERROR-Meldung. (Nun ja, das ist auch eine "Antwort"...)
 - Ende der Zwischenbemerkung.

Wie geht das vor sich?

Lade BASIC und gib ein: PRINT 3=4

BASIC antwortet hierauf leider nicht: "Gehe nochmal in die Schule und gib (!) dir eine Chance!" - sondern sagt schlicht: "NULL!" (0).

Gib jetzt ein: A = (3=4)

PRINT A

Die Antwort lautet wieder: "NULL!" (A=0)

Auf A=3>4

PRINT A

folgt ebenfalls: A=0

Nun aber: A = 3<4

Print A

---> -1 erklärt uns der fröhliche Bildschirm! Dies aber nicht etwa deshalb, weil 3-4=-1 (was ja in der Tat richtig ist), sondern weil es WAHR ist, was da geschrieben steht.

Wir fragen jetzt mal: PRINT 3 < 100. Auch hierauf lesen wir -1, obwohl 3-100=-97 ist (glaube ich).

Jetzt schreiben wir: $A = 3 * 12$
PRINT A

und erhalten 36 (ich fragte meinen Enkel. Es stimmt!! Der kann vielleicht rechnen!...) und nicht -1, denn "3*12" ist ja keine Behauptung, sondern eine Frage. Fragen beantwortet der Computer, so gut er kann (jawohl, das Kleine und auch das Große Einmaleins - das kann er!)

Nun fragen wir nicht, sondern wissen, daß $3*12=36$ ist. Mit diesem Wissen formulieren wir die Behauptung:

$$A = (3 * 12) = 36$$

Auf PRINT A antwortet BASIC: "Ja, stimmt!" - allerdings in seinem Jargon nur kurz: "-1".
Probiert's: $A = (3*12=35)$! Antwort: $A=0$ (in Worten: "Du lügst!")

Kurzum: Wenn wir eine Variable mit einer Gleichung oder auch einer Ungleichung laden, so nimmt sie nicht irgendeinen Zahlenwert daraus an, sondern "-1 XOR 0"; in Worten:

Entweder -1, wenn die (Un-)Gleichung WAHR,
oder 0, wenn sie NICHT-WAHR war.

Kleiner Alptraum:

Aus dem, was früher einmal wahr war,
ergibt sich später oftmals WIRWAR...

Auch eine falsche "Gleichung" ist nicht etwa "unlogisch", sondern sie ist die (oft absichtliche) "logische Darstellung von etwas Falschem"! - Alles klar?

(Nicht nur Diplomaten und Werbeleute argumentieren so spitzfindig! Mathematik dringt sehr spitz überall ein, und mathematische Logik hilft ihr beim Anspitzen...)

Das Tollste ist: Die gleichen Späßchen kann man nicht nur mit Zahlen (numerisch), sondern auch mit Strings veranstalten. Und das macht die "Steno-Logik" so reiz- und besonders wertvoll!

Eine praktische Anwendung dieser Zusammenhänge wäre etwa:
Wir besitzen ein Lexikon mit drei Bänden:

- Band 1: A - H
- Band 2: I - Q
- Band 3: R - Z

Wir suchen nun ein Wort: LINEINPUT "Was suchst du?"; W\$
Die folgende Zeile sagt uns, in welchem Band dieses Wort zu finden ist:

$$B\% = 1 - ((W\$ > "HZ") + (W\$ > "QZ")) \quad *)_{s.u.}$$

Wenn $W\$ = "FRANZ"$, so sind beide Ungleichungen NICHTWAHR (F ist nicht größer als H und erst recht nicht größer als Q). Die Klammern werden Null und es bleibt die 1, also Band 1! Ebenso für "HERMANN".

Wenn $W\$ = "LIESCHEN"$, ist die erste Ungleichung $W\$ > "HZ"$ WAHR = -1, die zweite aber 0 (NULL), da $L < QZ$. -1 von 1 abgezogen, ergibt 2 \Rightarrow Band 2! statt 2

Wenn $W\$ = "QZA"$, landen wir in Band 3 ($QZA > QZ$) und werden dort vergeblich nach dem Wort "QZA" suchen - weil es das nicht gibt. Wir werden es deshalb wohl auch kaum suchen!

Wenn $W\$ = "RUDI"$, dann treffen beide Ungleichungen zu und jede liefert -1. Somit erhalten wir: $B\% = 1 - (-1 + -1) = 3$! Im Band 3 unseres Nokixels finden wir garantiert "RUDI", falls er den Eintrag sich nicht verbat!



(Wer tät' das schon und ständ' nicht gern im Klexion??)



Ich habe das Z hinter H bzw. Q angefügt, damit man bei Eingabe von Wörtern, die mit H bzw. Q anfangen, nicht im nächsten Band landet. (Wieso? - Aha! Gelle? - Man kann das natürlich auch noch narrensicherer formulieren. Übt euch!) - Das Z hinter dem H bzw. Q mußte also sein!

Nun mag der eine odere der andere meinen, das sei doch etwas "umständlich" und er würde "Le mots" auch so finden (nicht von SARTRE), denn die Anfangsbuchstaben stünden ja auf den Buchrücken! Bittschön! Er begeben sich der Chance, "STENO-LOGIK" zu üben!

Oder er finde geistreichere und nützlichere Anwendungen! Das Rüstzeug dazu hat er nun. Und wenn diese wenigen Sätze zur Sache eine Anregung waren, so freut sich auch der Verfasser.

Aber wesentlich mehr, wie gesagt, bei ROSENFELDER!
Ihn empfiehlt euch ohnehin: **Xalot**

*) statt:

IF ((W\$ > "HZ") AND (W\$ < "QZ")) THEN B% = 2 ELSE
IF W\$ > "QZ" THEN B% = 3 ELSE B% = 1

Das ist 2,5 mal so lang!

PS: Solche (in abhängigen Variablen implizierten) Gleichungen bzw. Ungleichungen dürfen auch als Faktoren auftreten!

Kj

HEFT
16
November
1986

26

☞, ein UD-Zeichen auf dem Drucker

Die meisten von uns haben sehr intelligente Drucker, mit denen sich allerhand anstellen läßt. Dennoch wird gerne alles mögliche dem Textprogramm überlassen. So sind beispielsweise die Benutzer von TSCRIPS unfehlbar am Telefon zu erkennen, das wohl zum Lieferumfang des Programms für grafikfähige Drucker gehört. Einzelne von uns haben sich immerhin die Mühe gemacht, dem Telefon mit Hilfe der \$P-Option des Programms eine eigene Gestalt zu geben. Es geht auch anders.

Mit den meisten im Club vertretenen Druckern kann man einen eigenen Zeichensatz oder auch nur einzelne Zeichen selbst definieren (user defined characters, hier kurz UD-Zeichen genannt). Dazu wird der gerade angewählte Zeichensatz in ein User-RAM kopiert, anschließend kann man einzelne oder alle Zeichen mit eigenem Code modifizieren. Schließlich wird dieses User-RAM anstelle des EPROMs im Drucker selektiert. Nun steht der frei programmierte Zeichensatz zum Druck zur Verfügung.

Die Handbücher der Drucker geben sich alle Mühe, das Verfahren verständlich zu machen. Besser erklären kann ich es auch nicht. Aber vielleicht kann ich dem Leser Mut machen, es doch auch einmal zu versuchen. Es wäre wirklich schade, den teuren Drucker nicht voll auszureizen. Es lockt z. B. die Aussicht, auch ein eigenes Telefon für TSCRIPS zu kreieren und dabei nicht einmal einen Graphik-Zeichensatz laden zu müssen. Genau das habe ich auch getan und werde es jetzt beschreiben. Das Resultat war ein kleiner ☞.

Alles beginnt natürlich mit dem Entwurf des Zeichens. Die Abbildung zeigt den Telefonhörer (durchgezogene Linie) und den Versuch seiner digitalen Umsetzung (Punkte). Dabei sind in waagerechter Richtung je zwei Punkte für eine Linie vorgesehen, weil die Striche sonst zu dünn würden. Diese Punkte können nun als Bits etlicher Bytes aufgefaßt und in Hex oder wie auch immer notiert werden. Das Weitere ist von Drucker zu Drucker unterschiedlich. Hier wird der Vorgang für den NEC P6 erläutert.

Der P6 hat 24 Nadeln, die bei welcher Schrift auch immer, also auch bei UD-Zeichen, alle aktiviert werden. Deshalb sind zur Programmierung eines Zeichens in senkrechter Richtung pro Punktspalte drei Bytes erforderlich. In der Zeichnung trennen waagerechte Striche diese Drittel. Drucker mit 8 oder 9 Nadeln benötigen dafür nur ein Byte. Die Wertigkeit der Bits (Punkte) nimmt von oben nach unten ab, d. h. das höchstwertige Bit des jeweiligen Bytes entspricht dem oberen Punkt. Die Reihenfolge der drei Bytes ist oben-mitten-unten, die der Punktspalten geht von links nach rechts.

Die eigentliche Programmierung sollte mit einem Reset des Druckers beginnen, um etwa zuvor eingestellte und evtl. ungünstige Steuerungen zu löschen. Anschließend wird ein Zeichensatz selektiert, in diesem Fall die Proportionalsschrift. Bei anderen Druckern (z. B. Gemini-10X) erübrigt sich das. Da hier nur ein einziges Zeichen neu definiert werden soll, wird das User-RAM sodann mit dem gerade selektierten Zeichensatz angefüllt. Sonst wäre es nämlich am Ende bis auf das eine neue Zeichen vollkommen leer, jeder Buchstabe würde als Blank wiedergegeben.

Schließlich kommt der Befehl zur Vorbereitung des User-RAMs. Er beinhaltet eine bestimmte Steuersequenz und den ASCII-Code des zu ändernden Zeichens. Beim NEC werden die Ober- und Untergrenze des ASCII-Bereiches eingegeben, der neu definiert werden soll. Da es hier nur um ein Zeichen geht, in diesem Falle 7Fh, sind Ober- und Untergrenze identisch. Zuletzt muß dem Drucker noch mitgeteilt werden, wie breit (in Punkten) das Zeichen werden soll. In diesem konkreten Beispiel beginnt es mit 2 leeren Dotspalten, fährt fort mit 37 Spalten, die Punkte enthalten werden und einer wieder leeren Spalte. Auch dies ist beim Gemini nicht erforderlich;

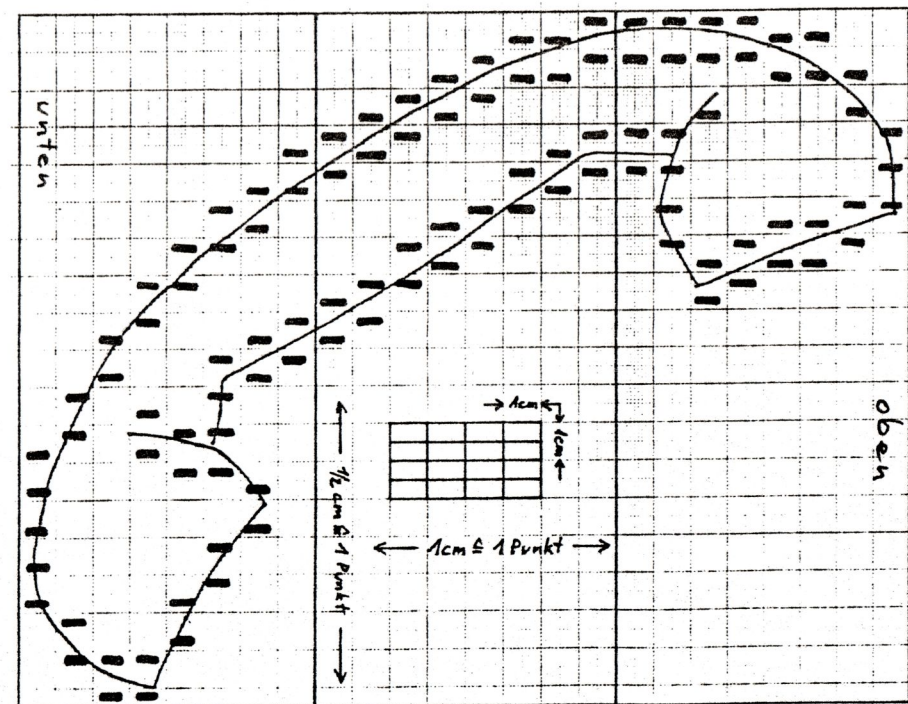
lest euch das im Druckerhandbuch einfach durch.

Nun folgen in der langen Tabelle die Codes für die 37 Punktspalten, für jede drei Bytes. Wenn man sich diese Hexzahlen binär vorstellt, erkennt man in der Zeichnung die einzelnen Bits als gesetzte Punkte wieder. Weniger aufwendige Schriftarten als LQ PS erfordern weniger Code. Beim Gemini kostet es, soweit ich mich erinnere, sogar nur ca. 25-30 Bytes insgesamt. Die Auflösung der Skizze muß diesen Gegebenheiten selbstverständlich Rechnung tragen. Also weniger malen.

Bisher operierten wir noch immer vom druckerinternen Zeichensatz aus. Der neu gestaltete wird nun selektiert. Am Ende der Tabelle erscheint nun das Zeichen selbst, das wir natürlich sogleich auf dem Papier bewundern wollen. Ein nachfolgender Wagenrücklauf (ODh) sorgt dafür, daß der Drucker sofort loslegt und nicht erst etliche Zeichen in seinem Puffer ansammelt.

Dieser ganze Klumpatsch von insgesamt 134 Bytes wird von einer kurzen Routine auf den Drucker ausgegeben, die beim Label start beginnt. Sie soll hier nicht näher erläutert werden. Wer einen Tandy hat, muß OUTI durch ein Kostrukt ersetzen, das anstelle des Ports FDh die Speicherstelle 37E8h bedient. Genauso einfach geht es in BASIC. Dazu verstaut man am besten alle erforderlichen Codes in DATA-Zeilen, die in einer READ-Schleife auf den Drucker ausgegeben werden.

Arnulf Sopp, ☞ 0451-791926



HEFT
16
November
1986

11


```

5200      00001      ORG      5200h
00002
5200 211052 00003 start LD      HL,telefon      ;Tabelle der Drucker codes
5203 01FD86 00004 LD      BC,tabend-telefon*256+0fdh ;B(-Länge
00005
5206 CDD105 00006 loop CALL    05d1h      ;C (- Druckerport FD
5209 20FB 00007 JR      NZ,loop      ;Druckerstatus lesen
520B EDA3 00008 OUTI      ;falls Dr. nicht bereit
520D C8 00009 RET      Z      ;Byte ausgeb., Zeig. erh.
520E 18F6 00010 JR      loop      ;ins DOS, wenn fertig
00011 ;sonst weiter ausgeben

5210 1C 00012 telefon DB      1ch,'s'      ;Reset für den Drucker
5212 1B 00013 DB      1bh,'p',01h      ;Schriftart LQ PS
5215 1B 00014 DB      1bh,':',00h,00h,00h ;copy PS-font
521A 1B 00015 DB      1bh,'&',00h      ;UD-Definition
521D 7F 00016 DB      7fh,7fh      ;ASCII 7F programmieren
521F 02 00017 DB      02h,25h,01h ;2 + 37 + 1 Dots breit
00018
5222 0F 00019 DB      0fh,80h,00h ;ab hier eigentl. Zeichen
5225 30 00020 DB      30h,60h,00h
5228 0F 00021 DB      0fh,90h,00h
522B 70 00022 DB      70h,68h,00h
522E 00 00023 DB      00h,14h,00h
5231 44 00024 DB      44h,0ah,00h
5234 83 00025 DB      83h,85h,00h
5237 00 00026 DB      00h,42h,80h
523A 83 00027 DB      83h,0a1h,00h
523D 00 00028 DB      00h,40h,0c0h
5240 C2 00029 DB      0c2h,30h,20h
5243 30 00030 DB      30h,08h,40h
5246 4A 00031 DB      4ah,14h,30h
5249 34 00032 DB      34h,08h,00h
524C 08 00033 DB      08h,06h,18h
524F 04 00034 DB      04h,01h,00h
5252 00 00035 DB      00h,02h,88h
5255 00 00036 DB      00h,01h,44h
5258 00 00037 DB      00h,00h,0a0h
525B 00 00038 DB      00h,00h,44h
525E 00 00039 DB      00h,00h,22h
5261 00 00040 DB      00h,00h,08h
5264 00 00041 DB      00h,00h,32h
5267 00 00042 DB      00h,00h,09h
526A 00 00043 DB      00h,00h,30h
526D 00 00044 DB      00h,00h,41h
5270 00 00045 DB      00h,00h,00h
5273 00 00046 DB      00h,00h,41h
5276 00 00047 DB      00h,00h,20h
5279 00 00048 DB      00h,00h,01h
527C 00 00049 DB      00h,00h,20h
527F 00 00050 DB      00h,00h,11h
5282 00 00051 DB      00h,00h,02h
5285 00 00052 DB      00h,00h,10h
5288 00 00053 DB      00h,00h,0eh
528B 00 00054 DB      00h,00h,00h
528E 00 00055 DB      00h,00h,0ch
00056
5291 1B 00057 DB      1bh,'i',01h ;UD-Zeichensatz selekt.
5294 7F 00058 DB      7fh,0dh ;Zeichen drucken und Ende
00059 tabend
00060
5200      00061      END      start      ;dort Einsprung

```

SYSCOPY1 und RAMDISK für Helmut Bernhardt's Banker

Ihr habt ja alle hoffentlich aufmerksam den Artikel über Helmut's Banker und das erste Programm dafür, RAMSYS von Arnulf, gelesen. Der Banker hat mich sofort fasziniert und so mußte er her. Leider stellte sich ein Problem auf dem Genie IIIs gibt es die Programme SYSCOPY und MEMDISK, die aber mit dem Banker nicht laufen. SYSCOPY leistet Ähnliches wie RAMSYS, nur benötigt es absolut keine Systemdiskette mehr. Alle SYS-Files lassen sich aus dem Speicher laden. MEMDISK versorgt den Benutzer mit einer 128K-Disk als höchstes Laufwerk (vorhandene Laufwerke +1). Diese Ramdisk hat durch ihre Geschwindigkeit unheimliche Vorteile gegenüber den lahmen Diskettenlaufwerken.

Ich habe mir also erlaubt, auch ohne TCS oder Phoenix um Erlaubnis zu bitten, diese beiden Programme zu disassemblieren und auf Helmut's Banker umzuwickeln. Das ist nicht so problematisch, wenn man viel Zeit aufbringt und auf einige Disketten verzichten kann, die man beim Probieren zerschneidet. Die Produkte heißen bei mir SYSCOPY1 und RAMDISK.

Im Gegensatz zu RAMSYS arbeitet SYSCOPY mit einem kleinen Directory, in das die SYS-Files beim Lesen eingetragen werden. Somit entfällt jegliche Änderung am Programm selber. Es dürfte mit allen Newdos/Gdos-Versionen laufen, solange SYS0 nicht stark verändert wurde, was kaum zu befürchten ist. Die SYS-Files liegen dann in Bank 1 und 2, wobei in Bank 2 je nach Länge/Anzahl der Files noch mehr oder minder viel Platz ist, den man später problemlos nutzen kann. In Bank 1 stehen auch die Service-Routinen. Nach dem Aufruf lädt SYSCOPY erst einige Sekunden seine SYS-Files und dann kann man die System-Diskette entfernen. Beim Kopieren wird zwar noch danach gefragt, aber es kann auch eine Pappscheibe in Drive 0 liegen, wenn die als Diskette anerkannt wird. Alle SYS-Files werden aus den Banks geladen.

RAMDISK benutzt einigen Platz in Bank 2, den SYSCOPY1 frei läßt. Als RAM-Disk dienen die Banks ab 3 aufwärts. Beim Aufruf kann man die RAM-Disk in 64K-Schritten (also jeweils 2 Banks) von 64K bis 896K einstellen (falls Euer Banker soweit bestückt ist). Wenn schon vorher einmal RAMDISK gestartet worden war (auch vor einem Reset), bleibt der alte RAM-Disk-Inhalt erhalten, wenn man nicht ausdrücklich eine Neu-Formatierung wünscht. Die RAM-Disk wird als Laufwerk 2 oder 3 angesprochen (beispielsweise läuft auch SUPERZAP!). Vor dem ersten Start muß man eine kleine Änderung am Source-Code vornehmen und das Programm neu assemblieren, weil jedes DOS (Newdos, Gdos, Hdos) einen anderen Initialisierungswert benötigt. Das dürfte aber nicht zu schwer sein.

SYSCOPY1 und RAMDISK sind über die Clubbibliothek des Club 80 erhältlich.

Gerald Schröder

* Mein 3. Assemblerprogramm *

Aller guten Dinge sind drei! - Also dürfte dies hier eigentlich nicht mehr von mir kommen! Aber es heißt ja:

Unter den Blinden ist der Einäugige König!

Womit ich nicht sagen will, daß ihr blind seid...

Vielmehr: daß ich nicht "zweiäugig" bin - was das auch immer heißen mag.

Kurzum: Für die "Sehenden" ist dies hier wiederum nicht geschrieben. Sondern für diejenigen, die gern mal "hinter die Kulissen" schielen möchten - falls sie nicht bereits zu den Drahtziehern gehören, die die Puppen tanzen lassen.

Dank schulde ich zweien, die mir beim Drahtziehen halfen: den Herren Schröder und Sopp (in alphabetischer Folge) oder auch den hilfreichen Kollegen Arnulf und Gerald (ebenefalls in alphabetischer Reihenfolge!)

Nicht, daß sie das nachstehende Programm für mich oder mit mir gebastelt hätten - das wäre zu einfach für mich gewesen! Es ist schon eine eigene Creation (wie man unschwer an seinem hochgeistigen Inhalt erkennt...). Aber sie erklärten mir in gewohnt selbstloser und hingebender Weise einige Tricks/Drehs/Wendungen-/Kniffe, die ich - wieder einmal! - in keinem Leerbuch fand!

Sicher ist auch dieses nun schon etwas längere Programm möglicherweise noch ungeschickt. Aber ich kann nur wieder sagen:

Es läuft!

Also ist es nicht falsch. Höchstens noch nicht rationell oder tricky.

Nach meinem ersten Programm (vorletztes INFO) mit fest vorgegebener MM (Bildschirm-Meldung oder auch "Monitor Message") und meinem zweiten (letztes INFO) mit MM (Mathematischer Musik oder auch Musikalischer Mathematik) widmet dieses dritte sich nun der vornehmsten Aufgabe eines Computers: mit seinem Herrchen - oft "USER" genannt, was wohl soviel heißt wie

"Unermüdliches Spielen Erzeugt Routine".

zu sprechen! Was auch als

Dialog

bezeichnet wird. Er ist zwar kurz und bündig. Aber ich hoffe, er zeigt denjenigen, die wie ich Anfänger/Einsteiger sind, wie man ihn grundsätzlich aufbaut. Lange, lange habe ich danach gesucht. Obgleich es nach meiner Meinung die wichtigste Aufgabe ist, die man einem Computer anvertrauen kann, steht dafür nirgends ein praktisches Beispiel - weder im berühmten R.ZAKS noch sonstwo (wer mir das Gegenteil mitteilt, sei begrüßt und bedankt!)

Immerhin, eine gute Hilfe war mir

Craig A. Lindley, "TRS-80/Z80 Assembly Language Library"

Wayne Green Publications 1983, ISBN 0-88006-060-3. (Nur rd. DM 130)

Das Programm greift auf die LDIR-Funktion sowie auf die Bedeutung des \$-Zeichens zurück; beides habe ich schon im ersten Beitrag erklärt. Im übrigen habe ich diesmal versucht, es besser zu kommentieren, nachdem mich Arnulf mit Recht gerügt, daß ihr ja nicht lachen, sondern lernen wolltet! Wahrscheinlich hat bisher "keiner weder gelacht noch gelernt"! (Ich hoffe, ihr erkennt die doppelte Verneinung, sonst bringe ich das nächste Mal einen Kursus über den Umgang mit logischen Operatoren. Bitte melden!)

Zwei Fehler aus den früheren Beiträgen habe ich zu korrigieren:

"OPCODE" und "OBJECT CODE" ist natürlich nicht dasselbe:

OPCODE = Operation Code = Befehlscode = der Anweisungsteil, mit dem jeder Befehl an die CPU beginnt; ihm folgt der "Operandenteil", d.i. der "Befehlsempfänger" (sozusagen der "Thumbe Thor", heute auch "G.I." oder "Sad Sack" genannt), der dem Befehl gehorchen muß (manchmal ist da auch gar keiner, der zu gehorchen hat; dann kommandiert der UVD sich selbst!)

Wenn sämtliche OPCODES+OPER(AND)IEREN, die den Source- (Quell)-Code darstellen, "assembliert" sind, haben wir endlich den OBJECT CODE, der nur noch aus Sedezimalzahlen besteht, die die verkürzte Form der eigentlichen BIT-Konfigurationen des Maschinencodes darstellen.

Und die zweite Korrektur:

Speicheradressen-Definitionen, die durch die Pseudo-Anweisungen (jetzt wissen wir ja, was das ist)

DEFB = Define Byte

DEFW = Define Word = 2 Bytes

DEFM = Define Message

DEFS = Define Storage (wie in vielen sogenannten Leerbüchern

zu lesen steht; ich bin aber mit Arnulf der Meinung,

daß die Erklärung "Define Space" logischer ist)

gesetzt werden, an denen das Definierte beginnt, dürfen

nie vor, und solche, die einen konstanten Wert (z.B. eine ROM-

Adresse) festlegen - wie die EQU-Gleichsetzungen - dürfen ni

ch hinter der ORG-Anweisung (auch eine Pseudo-Anweisung)

stehen! Aus Fehlern wird man klug (ich müßte infolgedessen inzwischen schon sehr klug sein; stimmt aber nicht: folglich habe ich

noch viele Fehler vor mir! Merkt euch das und seid vorsichtig!)

Das Wichtigste in diesem Musterprogramm sind die drei Zeilen 100, 110 und 130. Sie zeigen den Rückgriff auf ROM-Routinen!

Das englische Wort "idle" ist verwandt mit "eitel" und bedeutet

"faul". Je fauler der Programmierer ist, um so mehr überläßt er

die Arbeit den Routinen oder Unterprogrammen, die fertig im ROM

vorliegen ("gewußt, wo!" Haha!-Siehe "ROM-Listing" von Röckrath -

kommt Röckrath,

kommt ROM-Rat!

- und um so "eitler" gefällt er sich in diesem (Nichts-)Tun!

Natürlich ist das nicht eitel, sondern "edel".

Mindestens genau so edel ist der Zugriff auf DOS-Routinen. Manche meinen, noch edler. Hier hilft das DOS-Buch von Grosser

(bei Röckrath) weiter (das allerdings nur ohne den obligaten

Bindestrich zu haben ist, genau wie das "Maschinen Sprache Buch"

von Röckrath; Maschinensprache ist eben Maschinensprache, nicht

Deutsch, ganz klar!)

Doch hierüber ein andermal.

Vielleicht.

Für heute schließe ich. Egal:

Es reicht.

Kajol


```

00010 ; DIALOG MIT DEM ORAKEL ZU WEINHEIM
00030 ; VON KAJOT, GESTARTET 28.10.86
00040 ; KEINE RECHTE ANGEMELDET!
00045 ; NACHAHMUNG EMPFOHLEN !
00050 ;
00060 ; ZUERST WERDEN 3 ROM-ROUTINEN DEFINIERT:
00070 ;
01C9 00100 CLS EQU 01C9H ; CLEAR SCREEN
0049 00110 INCHW EQU 0049H ; 1 ZEICHEN EINGEBEN!
05D9 00130 INBUFF EQU 05D9H ; 1 ZEILE EINGEBEN!
00133 ;
6000 00136 ORG 6000H ; HIER SPEICHER-ANFANG
00140 ;
00150 ; DEFINITION VON CODE-WOERTERN = SPEICHERADRESSEN
00160 ;
6000 57 00170 FRAGE1 DEFM 'WILLST DU VERREISEN <J/N> ?'
601B 4F 00210 ANTW1 DEFM 'OK - BLEIBE HIER UND ASSEMBLIERE!'
603C 57 00230 FRAGE2 DEFM 'WOHIN?'
6042 42 00270 ANTW2 DEFM 'BLEIB LIEBER ZUHAUS UND LERNE WAS!'
00290 ;
00300 ; NACH DIESEN VORBEREITUNGEN BEGINNT
00305 ; DAS EIGENTLICHE PROGRAMM
00310 ;
6064 CDC901 00330 START CALL CLS ; KLAR SCHIFF!
6067 011B00 00340 LD BC,ANTW1-FRAGE1; LAENGE DER 1.FRAGE
606A 114A3C 00350 LD DE,3C4AH; SIE SOLL IN DIE 2.ZEILE
606D 210060 00360 LD HL,FRAGE1; ADRESSE, WO DIE FRAGE
00370 ; BEGINNT, WIRD GELADEN
6070 EDB0 00380 LDIR ; TRANSPORT DER FRAGE
00390 ; ZUM BILDSCHIRM
00400 ;
00410 ; DIE ANTWORT <J ODER N> WIRD ERWARTET,
00420 ; BEI VERNEINUNG FOLGT VERABSCHIEDUNG,
00430 ; BEI BEJAHRUNG ERFOLGT WEITERE FRAGE -
00440 ; NUR MIT GROSSBUCHSTABEN, ALSO
00450 ; "UPPER CASE" =APPEN(ZELLER) KAS ANTWORTEN!
00460 ;
6072 CD4900 00470 ANTWJN CALL INCHW ;EQU SIEHE OBEN!
6075 CBAF 00480 RES 5,A ;BIT 5 ZURUECKSETZEN, FALLS
00490 ;MIT KLEINBUCHSTABEN (LC)
00500 ;GEANTWORTET WURDE
00510 CP 'J' ; LAUTETE DIE ANTWORT 'J'?
00520 JR Z,FRAGE ;FALLS JA, ZUR 2.FRAGE!
00530 CP 'N' ; ODER LAUTETE SIE 'N'?
00540 JR NZ,ANTWJN; FALLS AUCH NICHT, DANN
00550 ; WIEDERHOLE DIE ANTWORT!
00560 LD BC,FRAGE2-ANTW1; FALLS NEIN, FOLGEN
00562 ; ABSCHIEDSWORTE, DEREN LAENGE
00564 ; ERGIBT SICH AUS DER DIFFERENZ
00566 ; DER ZEILEN 'FRAGE2' - 'ANTW1'
00570 LD DE,3C8AH; SIE STEHN GENAU DARUNTER
00580 LD HL,ANTW1; SIEHE DIE MESSAGE OBEN
00590 LDIR ; LDIR SCHREIBT'S HIN
00595 CALL INCHW ; AUSSTIEG MIT ENTER!
00600 JP 402DH ; ZURUECK INS DOS - ENDE!
00610 ;
00620 ; FALLS DU VERREISEN WILLST, WIRST DU GEFRAGT:
00625 ; 'WOHIN?'
00630 ;
6090 010600 00640 FRAGE LD BC,ANTW2-FRAGE2; LAENGE VON 'WOHIN?'
00645 ; ERKLAERUNG ANALOG ZEILE 564!
6093 118A3C 00650 LD DE,3C8AH ; SIE SOLL IN DIE 3.ZEILE
6096 213C60 00660 LD HL,FRAGE2; "WOHIN?"
6099 EDB0 00670 LDIR ; LDIR SCHREIBT'S HIN!

```

```

00680 ;
00690 ; NUN ANTWORTE! ES WERDEN HIERFUEER 48 STELLEN
00700 ; ZUR VERFUEGUNG GESTELLT (D.I.DER REST DER ZEILE)
00710 ;
609B 0630 00720 LD B,30H ; 30H = STELLENZAHL
609D 21913C 00723 LD HL,3C91H; CURSOR-POSITION FUEER
60A0 222040 00726 LD (4020H),HL ; DIE ANTWORT
60A3 210042 00730 LD HL,4200H; PUFFER F.D.ANTWORT
60A6 CDD905 00740 CALL INBUFF ; EQU SIEHE OBEN
00750 ;
00760 ; DAS ORAKEL ZU WEINHEIM ANTWORTET:
00770 ;
60A9 012200 00780 LD BC,START-ANTW2; LAENGE DER ANTWORT
60AC 110A3D 00790 LD DE,3D0AH; ZEIG SIE 2 ZEILEN TIEFER
60AF 214260 00800 LD HL,ANTW2; SIEHE MESSAGE OBEN
60B2 EDB0 00810 LDIR ; ZUM BILDSCHIRM
60B4 CD4900 00815 CALL INCHW ; AUSSTIEG MIT ENTER!
60B7 C9 00820 RET ; ZURUECK ZUM AUFRUFER!
6064 00850 END START
00000 TOTAL ERRORS
32432 TEXT AREA BYTES LEFT

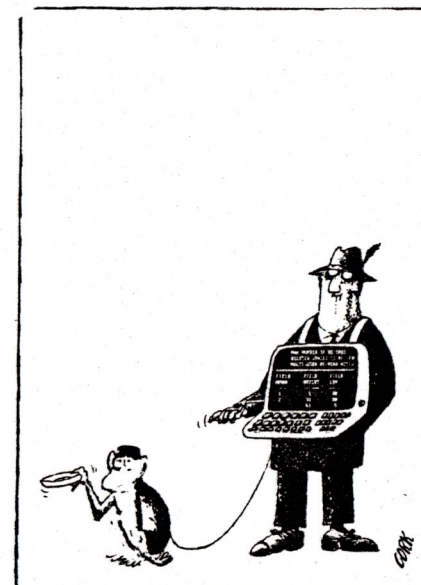
```

```

ANTW1 601B 00210 00340 00560 00580
ANTW2 6042 00270 00640 00780 00800
ANTWJN 6072 00470 00540
CLS 01C9 00100 00330
FRAGE 6090 00640 00520
FRAGE1 6000 00170 00340 00360
FRAGE2 603C 00230 00560 00640 00660
INBUFF 05D9 00130 00740
INCHW 0049 00110 00470 00595 00815
START 6064 00330 00780 00850

```

W.J.



Liebe Clubfreunde,
hier folgt der Rest der RPML-Beschreibung. Die nachstehende
Liste enthält eine nach Funktionen sortierte Aufstellung aller
in RPML vorkommenden Anweisungen.

Herb Hill

* Summary of Commands *

RPML Vers. 2.0 1986

In dieser Kurzanleitung werden für Daten folgende Abkürzun-
gen benutzt:

n für eine 16bit Zahl
d für eine 8bit Zahl
adr für eine 16bit Adresse
flag für logisches Flag
name für eine 16bit Buffer-Adresse

Logische Befehle:

XOR	(n1 n2	-> n3) Logisches XOR
OR	(n1 n2	-> n3) Logisches OR
AND	(n1 n2	-> n3) Logisches AND
NOT	(n1	-> -n1) Logische Negation
TRUE	(-> flag) Liefert 0FFFFH als Wahr-Flag
FALSE	(-> flag) Liefert 0000H als Falsch-Flag

=	(n1 n2	-> flag) Vergleich zweier Zahlen
=	(n1	-> flag) Test auf n1 gleich 0
<	(n1 n2	-> flag) Kleiner als
<=	(n1 n2	-> flag) Kleiner gleich
>	(n1 n2	-> flag) Größer
>=	(n1 n2	-> flag) Größer gleich
<>	(n1 n2	-> flag) Ungleich

Arithmetische Befehle:

+	(n1 n2	-> n3) Addition zweier Zahlen
-	(n1 n2	-> n3) Subtraktion zweier Zahlen
/	(n1 n2	-> n3) Division zweier Zahlen
*	(n1 n2	-> n3) Multiplikation zweier Zahlen
MOD	(n1 n2	-> n3) Liefert den Divisionsrest
/MOD	(n1 n2	-> n3 n4) Liefert das Divisionsergebnis und den Rest (n4)
RND	(-> n1) Liefert eine Zufallszahl zwischen 0...65535

Stack-Befehle:

DROP	(n1	->) Vernichtet den TOS
SWAP	(n1 n2	-> n2 n1) Vertauscht TOS & NOS
OVER	(n1 n2	-> n1 n2 n1) Copiert NOS zum TOS
ROT	(n1 n2 n3	-> n2 n3 n1) Rotiert n1 zum TOS
INC	(n1	-> n1+1) Erhöht TOS um 1
DEC	(n1	-> n1-1) Erniedrigt TOS um 1

Zuweisungs-/Lade-Befehle:

:=	(n1 adr	->) Legt n1 unter adr ab
:= (B)	(d1 adr	->) Legt d1 unter adr ab
?	(adr	-> n1) Holt den Wert von adr
? (B)	(adr	-> d1) Holt Byte von adr

I/O-Befehle:

INCHAR	(-> d1) Holt ein Zeichen von der Tastatur (ohne zu warten !)
READ(S)	(adr	->) liest einen String nach adr von d. Tastatur
READ(CH)	(adr	->) Liest einen String nach adr von d. Tastatur
OUTCHAR	(d1	->) Gibt ein Zeichen aus
PRINT	(adr	->) Gibt die unter adr befindliche Zahl aus
PRINT(S)	(adr	->) Gibt den unter adr befindlichen String aus

I/O-Steuerbefehle:

SCROFF	(->) Schaltet die Bildschirm- ausgabe ab
SCRON	(->) Schaltet die Bildschirm- ausgabe wieder ein
DROFF	(->) Schaltet die Druckeraus- gabe ab
DRON	(->) Schaltet die Druckeraus- gabe ein
CLS	(->) Löscht den Bildschirm
PAGE	(->) erzeugt einen Seitenvor- schub auf dem Drucker
CR	(->) Erzeugt einen Zeilenvor- schub

System-Variable:

SYSFIELD	(-> adr) Liefert adr auf den ersten freien Speicherplatz am Programmende
ZEICHEN	(->) Enthält das Zeichen nach einem INCHAR-Befehl

Disk-Befehle:

```

READF      (name -> n1 flag) Liest ein byte von d. Disk
WRITEF     (n1 name -> flag) Schreibt ein byte auf Disk
OPEN(I)    (name -> flag) Eröffnet File zum Lesen
OPEN(O)    (name -> flag) Eröffnet File zum Schreiben
OPEN(R)    (name -> flag) Eröffnet Schreib/Lese-File
CLOSE      (name -> ) Schließt einen File
DELETE     (name -> ) Löscht einen eröffneten File
EOF?       (name -> flag) Prüft auf END-OF-FILE
FILESPEC   (adr name -> flag) Weist einem geschlossenen File
                               eine neue Filespec zu

```

System-Befehle:

```

NEW      (      -> ) Löscht das Programm u. initiali-
                               siert den Compiler neu
SYSTEM   (      -> ) Zurück zum DOS
NOFILES  (      -> ) Löscht alle Diskbefehle aus
                               dem Runtime-Modul
LIST     (      -> ) Listet alle bisher comp. Worte
MEMORY   (      -> ) Liefert den freien Speicherplatz
SAVE     (filespec->) Speichert das Programm auf Disk
COMPILE  (filespec->) Compiliert ein Programm
FILES    (      -> ) Listet das Directory

```

Compiler-Anweisungen:

```

CALL      Leitet Compilierung von Hex-Inlinecode ein für Unter-
           Programme
CEND      Beendet CALL-Anweisung
CODE      Leitet Compilierung von Hex-Inlinecode für in
           Assembler definierte RPNL-Worte ein
CEND      Beendet CODE-Anweisung
DECLARE   Eröffnet die Compilierung von Variablen, Konstanten,
           und dergleichen
DEND      Beendet DECLARE-Anweisung
PROGRAM   Leitet die Compilierung von High-Level Code ein
END       Beendet die PROGRAM-Anweisung

```

Schleifen-Strukturen:

```

FOR      (n2 n1 -> ) Initialisiert die FOR-Loop
1        (      -> n1 ) liefert den Laufindex der
                               aktuellen. FOR-Loop auf den
                               Stack
LEAVE    (      -> ) Bewirkt vorzeitigen Abruch
                               der aktuellen FOR-Loop
LOOP     (      -> ) Schließt die FOR-Loop ab
REPEAT   (      -> ) Beginn der Unbedingten Wieder-
                               holungsschleife
UNTIL    (flag -> ) Prüft, ob Aussprungsbedingung
                               erfüllt ist
LOOP     (      -> ) Schließt die REPEAT-Loop

```

Kontroll-Strukturen:

```

IF      (flag -> )
ELSE    (      -> )
ENDIF   (      -> )

```

```

) Beginn der Bedingungsanweisung
) Ende des TRUE-Pfades
) Ende der Bedingungsanweisung
und/oder des TRUE/FALSE-Pfades

```

TANDY ... better again

Unter diesem Motto stand eine Pressekonferenz am 5.9.86 in München, in der TANDY seine neuen Rechner vorstellte. Diese sind nun auch voll auf den Markt-Standard der Kompatibilität mit IBM eingestellt.

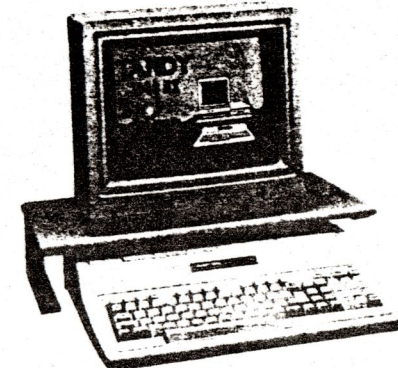
U. Vanden Bosche, TANDY Marketing-Direktor für Europa, stellte erhöhte Anstrengungen in Aussicht, auch in Europa den Verkauf von IBM kompatiblen Rechnern zu steigern, nachdem TANDY in den USA zum größten Hersteller von IBM kompatiblen Rechnern geworden ist. Um dieses Ziel zu erreichen, sollen die TANDY Computer Shops modernisiert werden und es soll eine umfassende Schulung der Mitarbeiter erfolgen.

Das Flaggschiff der neuen Serie ist der 3000 HD 40. Die

technischen Daten:

- INTEL 80286 CPU
- 8 MHz Takt
- 10 Erweiterungsplätze
- 3 für 8-bit XT
- 7 für 16-bit AT
- MS DOS 3.2 oder auf Wunsch XENIX, System 5
- 1 5/4 Zoll Laufwerk mit 1.2 MB Kapazität
- 1 40 MB Harddisk.

Dieser Rechner ist für Mehrplatzsysteme geeignet und TANDY wird auch in Kürze ein eigenes LAN, genannt ViaNet auf den Markt bringen.



1000 EX

Ein weiteres Mitglied der Rechnerfamilie ist der 3000 HL. Dies ist ein XT mit AT Qualitäten. Auch er besitzt eine 80286 CPU mit 8 MHz. Der Speicher ist durch Zusatzplatinen auf 4 MB aufrüstbar. Die anderen Daten entsprechen denen eines XT.

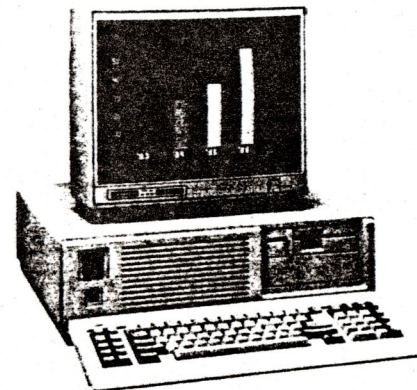
Der 1000 SX ist XT kompatibel, seine Taktfrequenz kann aber zwischen 4.77 MHz und 7.1 MHz umgeschaltet werden. Damit wird eine höhere Rechenleistung erzielt.

Nach unten wird die Palette mit dem 1000 EX abgerundet. Dieser Rechner ist von der Software IBM kompatibel, es wird MS DOS 2.11 verwendet. Um aber eine kompaktere Bauweise, und somit ein kleineres, handlicheres Gerät zu erzielen, ist die Hardware nicht IBM kompatibel.

Die technischen Daten dieses Rechners sind:

- 8088 CPU
- 4.77 oder 7.1 MHz Takt
- 256k, ausbaubar bis 640k
- MS DOS 2.11

Dieser Rechner wird für 1.750,- DM incl. Mehrwertsteuer, einschließlich einem s/w-Monitor angeboten. Natürlich besitzen alle Rechner eine deutsche Tastatur und einen parallelen Druckerausgang. Mit jedem Rechner wird das Programmpaket DeskMate II mitgeliefert, in das eine Textverarbeitung, eine Tabellenkalkulation, eine Datenbank, sowie eine Mailbox und ein Kalender integriert sind.



3000 HD 40

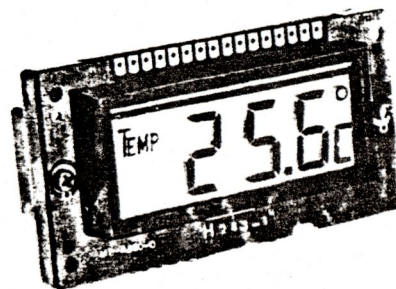
Hardwaretips: Digitales Temperatur- Zeitmodul

Nicht vorenthalten wollte ich euch das Datenblatt eines von mir auf der Hobbytronik erworbenen Uhrenmoduls, von dem ich glaube, daß es für alle Clubmitglieder interessant ist.

Das Modul ist sehr universell einsetzbar (Heizungsregelung, Innen/Außenthermometer incl. Uhr für Heim und Auto, Eiswarner usw.) und kann sogar die Temperatur, über eine serielle Schnittstelle, an Computer übergeben (ob das auch mit der Uhrzeit funktioniert habe ich noch nicht ausprobiert).

Zweieundzwanzig Märker kostete das Wunderdingchen auf dem CONRAD-Messestand, viel "teurer" kann es wohl auch im Versand nicht sein.

Hartmut Obermann



**Digitales
Temperatur-Zeitmodul**
Best.-Nr. 19 55 88

CONRAD

Technische Daten:

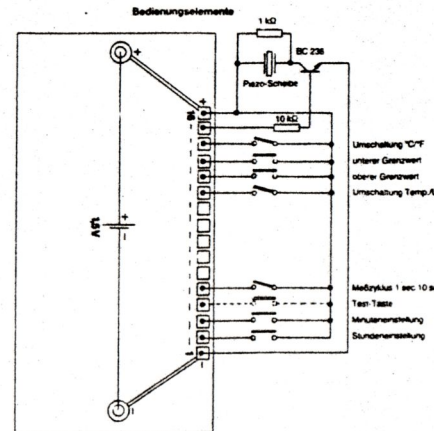
Meßbereich:	-20°C bis +70°C (0°F bis 160°F)
Auflösung:	0.1°C (F)
Genauigkeit:	±1°C (-20°C bis +70°C)
Meßrate:	10 Sekunden oder 1 Sekunde
Alarmausgang:	Intervalle 4 kHz - Dauer 6 Sekunden
Schaltausgang:	Oberer Grenzwert - Dauer 1 Minute Unterer Grenzwert - Dauer 1 Minute Oberer/unterer Grenzwert - Dauer 1 Sek.
Grenzwerteinstellung:	1°C-Schritte
Uhrenbetrieb:	12-Stunden-Anzeige
Betriebsspannung:	1,5 V (1,25 - 1,65 V)
Stromaufnahme:	ca. 3 µA

Externe Beschaltung:

Anschlußbelegung
 1: Masse (Minuspole der Versorgungsspannung)
 2: Stundeneinstellung
 3: Minuteneinstellung
 4: Testbetrieb (alle Segmente des Displays werden aktiviert)
 5: Meßraten-Umschaltung - Bei geschlossenem Schalter wird die Temperatur im Abstand von 1 Sekunde und bei offenem Schalter alle 10 Sekunden ermittelt. Beachten Sie, daß die Grenzwerteinstellung nur im 10-Sekunden-Betrieb möglich ist.
 6: Schaltausgang für den oberen Grenzwert. An diesem Ausgang liegt für die Dauer der Grenzwertüberschreitung - mindestens aber 1 Minute - „high“-Pegel (+1,5 V) an.
 7: Schaltausgang für den unteren Grenzwert. An diesem Ausgang liegt für die Dauer der Grenzwertunterschreitung - mindestens aber für 1 Minute - „high“-Pegel (+1,5 V) an.
 8: Steuerausgang für den oberen und unteren Grenzwert. An diesem Ausgang liegt bei Erreichen des oberen oder unteren Grenzwertes für die Dauer einer Sekunde „high“-Pegel an.
 9/10: Serielles Interface - Pin 9 ist der Datenausgang, an Pin 10 liegen die Clock-Impulse an.

CONRAD

- 11: Temperatur-Uhrzeit-Umschaltung: Bei offenem Schalter ist das Modul im Temperaturmodus, während bei geschlossenem Schalter die Uhrzeit angezeigt wird.
 12: Unterer Grenzwert
 13: Oberer Grenzwert
 14: Umschaltung °C oder °F: Bei offenem Schalter erfolgt die Temperaturanzeige in °Celsius, während bei geschlossenem Schalter °Fahrenheit angezeigt werden.
 15: Alarmausgang: Bei Grenzwertüberschreitung liegt an diesem Ausgang für die Dauer von 6 Sekunden ein Intervallton mit einer Frequenz von 4 kHz an.
 16: Betriebsspannungsanschluß (+1,5 V)



CONRAD

Bedienung:

Uhrzeiteinstellung:

1. Schalten Sie das Modul auf Uhrzeitanzeige (Schalter an Pin 11 geschlossen).
2. Drücken Sie die Minutentaste (Pin 3) für ca. 2 Sekunden bis der Doppelpunkt nicht mehr blinkt. Bei jedem Tastendruck wird die Uhr um eine Minute vorgestellt. Andauerndes Drücken bewirkt ein schnelles Verstellen mit einer Frequenz von 8 Hz.
3. Das Einstellen der Stunden erfolgt in gleicher Weise durch Betätigung der an Pin 2 angeschlossenen Taste.

Temperaturbetrieb:

Temperaturbetrieb:

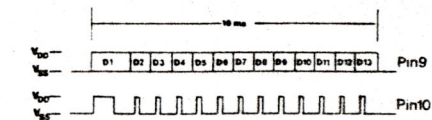
1. Schalten Sie das Modul auf Temperaturbetrieb (Schalter an Pin 11 offen).
 2. Schalten Sie auf Meßrate 10 Sekunden (Schalter an Pin 4 offen).
 3. Drücken Sie die an Pin 13 angeschlossene Taste. Im Display erscheint der untere Grenzwert. Drücken Sie gleichzeitig die Minutentaste, um den unteren Grenzwert einzustellen.
 4. Die Einstellung des oberen Grenzwertes erfolgt in gleicher Weise durch Betätigung der an Pin 12 angeschlossenen Taste in Verbindung mit der Minuteneinstelltaste.
 5. Bei aktiviertem Grenzwert erscheint Hi und (oder) Lo in der Anzeige. Der Grenzwertalarm kann unabhängig für Hi oder Lo stillgelegt werden, indem Sie gleichzeitig die entsprechende Grenzwerttaste und die Stundentaste drücken. Das Symbol (Hi oder Lo) wird dann im Display ausgeblendet.
- Beachten Sie, daß die Grenzwerteinstellung nur möglich ist, wenn das Modul auf Temperaturanzeige mit 10-Sekunden-Abtastrate geschaltet ist. Die Alarmausgänge funktionieren bei beiden Abtastraten (1 Sek/10 Sek), sie funktionieren nicht, wenn das Modul auf Uhrenbetrieb geschaltet ist.

CONRAD

CLUB 88 HARD
CLUB 88 HARD

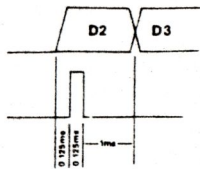
HEFT
26
November
1986

Serieller Datenausgang (Pin 9/10):



Bit-Tabelle

D _i	Hunderterteile bei Fahrenheitanzeige, Vorzeichen bei Celsiusanzeige (Minus = high, Plus = low)
D ₇	8
D ₆	4
D ₅	2
D ₄	1
D ₃	8
D ₂	4
D ₁	2
D ₀	1
D ₇	8
D ₆	4
D ₅	2
D ₄	1



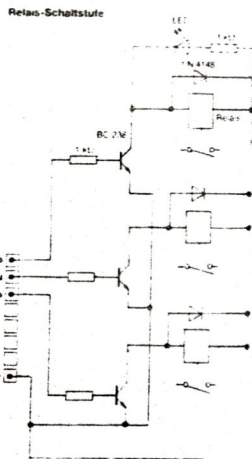
Alarm- und Schaltausgänge:

Alarmausgang: An Pin 15 liegt ein Alarmsignal an, welches bei Erreichen des oberen oder unteren Grenzwertes aktiviert wird. Das Signal steht für die Dauer von 6 Sekunden an und besteht aus einem Intervall mit einer Frequenz von 4 kHz. Über einen Schalttransistor kann sehr einfach ein piezokeramischer Schallwandler oder ein Miniaturlautsprecher (Kopfhörerkapsel 32 Ω) angesteuert werden.

Schaltausgang: An den Anschlüssen 6 und 7 liegt „high“-Pegel an, wenn die Temperatur den oberen oder unteren Grenzwert erreicht. Die Ausgänge sind solange „high“, wie die Grenzwertüberschreitung andauert, mindestens aber für 1 Minute. Beachten Sie, daß diese Mindestschaltzeit auch erhalten bleibt, wenn die tatsäch-

CONRAD

liche Temperatur bereits den entgegengesetzten Grenzwert erreicht hat. Die Schaltzustandsänderung erfolgt erst nach Ablauf dieser Periode. Am Modulanschluß 8 liegt „high“-Pegel an, wenn der untere oder obere Grenzwert erreicht wird. Im Gegensatz zu vorher beträgt die Dauer nur ca. 1 Sekunde.

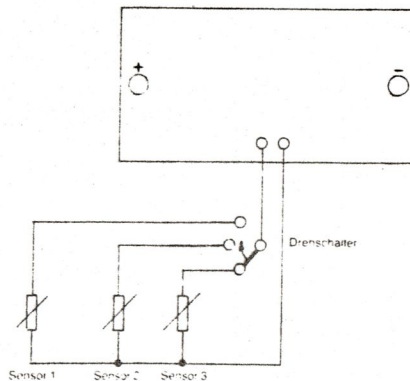


CONRAD

Meßstellenumschaltung:

Mit einem einpoligem Stufenschalter mit entsprechender Stufenzahl können beliebig viele Meßstellen angewählt werden. Pro Meßstelle ist ein eigener Sensor (Best.-Nr. 19 55 96) notwendig. Die Polarität der einzelnen Sensoren braucht nicht beachtet werden. Ebenso ist die Leitungslänge nicht kritisch. Der Leitungswiderstand darf jedoch 30 Ω nicht übersteigen. Bei ungünstiger Leitungsführung (parallel zum Lichtnetz) muß als Sensorleitung ein abgeschirmtes Kabel verwendet werden. Die Abschirmung wird dann mit Masse (Pin 1) verbunden.

Schaltbeispiel für drei Meßstellen



CONRAD

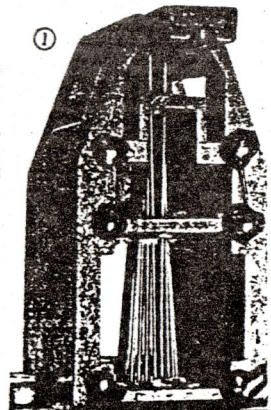
Computerlexikon

DRUCKER

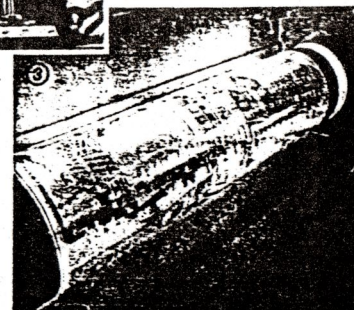
Ob sie nun hämmern, nadeln, spucken oder mit einem Laserstrahl schreiben, das ist mehr eine Frage der Schönschreibensprüche an einen Drucker. Dagegen ist der Drucker selbst nahezu ein Muß für jeden Computerbesitzer. Will man den Inhalt eines Bildschirms oder gar eine ganze Datei zu Papier bringen, wird ein Drucker unerlässlich. Wenn die Daten schon im Computer stecken, wäre es ja ein Wahnsinn, nun vom Bildschirm alles mit einer Schreibmaschine abzutippen. Das kann der Computer besser und schneller, ohne daß sich neue Fehler einschleichen. Die Daten zwischen Computer und Drucker werden dann seriell oder parallel übertragen. Bei der seriellen Übertragung »gehen« die Bits vom Computer zum Drucker über

eine Leitung wie im Gänsemarsch. Bei der schnelleren parallelen Übertragung stehen dagegen acht Leitungen zur Verfügung.

Gleichzeitig werden acht Bit nebeneinander übertragen, was einem Byte, also einem ASCII-Zeichen entspricht. Dann raten die Typenräder oder Kugelpföpfe wie bei einer Schreibmaschine los. Bei Nadeldruckern dagegen wird jeder Buchstabe aus Punkten zusammengesetzt, die das Nadelpaket des Druckkopfes blitzschnell zusammenstellt. Ebenfalls aus Punkten zusammengesetzt werden die Buchstaben bei einem Tintenstrahldrucker. Hier spuckt ein Satz Düsen winzige Farbtropfen gezielt aufs Papier. Neueste und schnellste Gruppe unter den Druckern sind die Laserprinter. Lautlos und fast mit Druckqualität spucken sie bis zu acht vollgeschriebene Seiten pro Minute aus.



②	50 μ s
	100 μ s
	150 μ s
	200 μ s
	250 μ s
	300 μ s
	350 μ s



Nadeldrucker
(1): Ein Bündel Stifte punktet die Buchstaben.
Tintenstrahldrucker (2): Eine Düse schleudert Tintentropfen.
Walze eines Laserdruckers (3): Sie ähnelt der eines Fotokopierers.

Technische Änderungen vorbehalten!

Alle Rechte, auch Übersetzungen, vorbehalten. Reproduktionen gleich welcher Art, ob Fotokopie, Mikrofilm oder Erfassung in Datenverarbeitungsanlagen, nur mit schriftlicher Genehmigung durch CONRAD ELECTRONIC GmbH.
© Copyright 1985 by CONRAD ELECTRONIC GmbH 8452 Hirschau

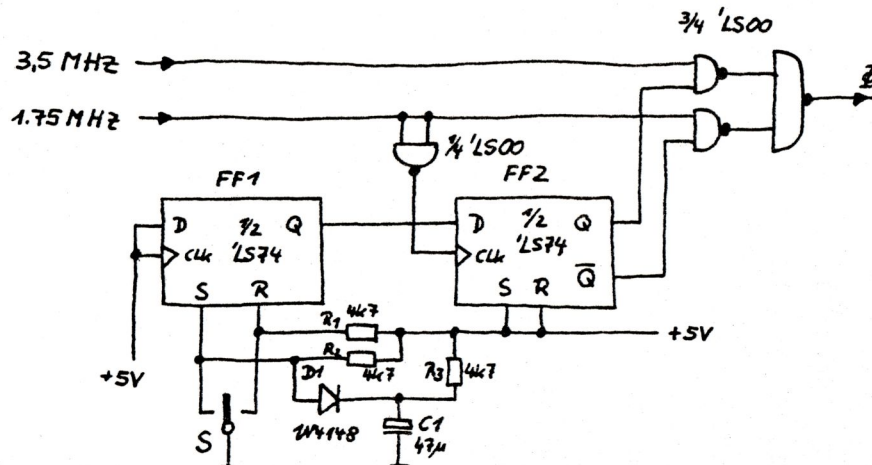
CONRAD

Taktumschaltung Die allerletzte Lösung

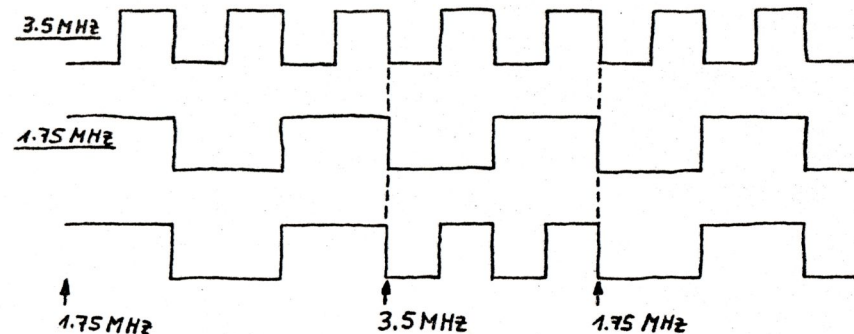
Wer seinen Rechner bereits mit allerlei nützlichen Dingen vollgestopft hat, der wird es kaum begrüßen, den ohnehin recht ansehnlichen Kabelverhau unnütz vergrößern zu müssen. Was also tun? Zuerst einmal ist festzustellen, welches "Kabel-Minimum" überhaupt notwendig ist. Da sind:

2x Stromversorgung
3x Taktsignal (2 Eing., 1 Ausg.)
3x Umschaltung
8x Anschlußleitungen

Vorrausgesetzt habe ich dabei, daß eine softwaremäßige Umschaltung nicht benötigt wird. Sonst ist ein Dekoder mit zusätzlichen 10 Leitungen nicht zu umgehen. Eben jenes aber galt es zu vermeiden. Damit ist der "Kupferbedarf" bestimmt. Blicke noch der "Siliziumaufwand". Die ideale Lösung besteht aus einem IC, nur die habe ich nicht gefunden. Macht auch nichts, mit zwei IC's geht's schließlich auch. Von Nöten sind also ein 74LS00 sowie ein 74LS74. Die NAND's wandern in einen Multiplexer, die Flip-Flops bilden einen Pulssynchronisierer. Das alles inklusive einiger Zutaten wie Lötzinn, Draht, Widerständen etc. in einen Topf, kurz aufkochen und dann in etwa einer 1/4 Stunde zusammenbauen. Für die allzeit bereiten Hobbyköche, die ihren Tauchsieder - sprich LötKolben - stets parat haben, hier nun das Kochrezept (=Schaltbild):



Die Schaltung baut darauf auf, daß die beiden CPU-Takte mittels Teilung durch 2 entstanden sind. Damit gibt es einen Zeitpunkt, zu dem beide Signale flankensynchron den Zustand wechseln. Oder anders ausgedrückt, beide Signale wechseln gemeinsam von HIGH nach LOW bzw. von LOW nach HIGH. Der ganze Witz der Schaltung basiert genau auf dieser Sache. FF1 wird nun mit einem Wechseltaster (Taster mit zwei möglichen Tastrichtungen) über S und R angesteuert (Gleichzeitig erfolgt so eine Entprellung des Tasters). Durch die Zwischenspeicherung des "Umschaltbefehls" in FF1 ist es nun möglich, den Taktwechsel zum richtigen Zeitpunkt vorzunehmen. Dieser kommt mit dem Auftreten der negativen Flanke des 1.75MHz Taktes. FF2 übernimmt dann die Information von FF1 und schaltet mit seinen Ausgängen den Multiplexer um. Nachfolgend hierzu ein Pulsdiagramm mit den beiden Takten und einer Umschaltsequenz SLOW -> HIGH -> SLOW. Der Inverter in der Taktleitung von FF2 ist notwendig, da sonst auf der positiven Flanke umgeschaltet wird.



Wie zu erkennen ist, verlängert/verkürzt sich jeweils eine Hälfte der betroffenen Periode. Das stört die CPU jedoch nicht weiter. Die Schaltung arbeitet so sicher, daß selbst schnelles hin- und herschalten zwischen den beiden Takten zu keinem "hängenbleiben" des 280 führt (nicht führen kann, da ja immer auf die negative Flanke zum Umschalten gewartet werden muß!).

Damit bei jedem Einschalten des Rechners stets der schnelle Takt angewählt ist, ist über die Diode D1, C1 und R3 ein Power-On-Reset eingebaut.

Somit steht nun eine weitere Variante der Taktumschaltung zur Wahl. Allen die derartiges noch nicht oder erst noch einbauen wollen, eine angenehme Wahl der Qual.

Tschüs

Reinhold Müller

*** * V e r k a u f * ***

 Model III (deutsche Tastatur, 2 Laufwerke)
 HRG (640 * 240) von TANDY
 absoluter Topzustand
 Preis: Gegen Gebot
 Thomas Buskowiak
 Tel.: 069 - 56 01 621

STAR Gemini-10X zu verkaufen
 Centronics-Interface (parallel; serielles Interface nachrüstbar), Übertragungsgeschwindigkeit 6000 cps, Druckgeschw. 120 cps
 8 intern. Zeichensätze, selbstdefinierte Zeichen, HRG, 6 versch. Zeichenbreiten, Sub- und Superscript, kursiv, versch. Schriftstärken, horizontal und vert. Tabulation usw.
 Bei Bedarf mit Druckerkabel (kostet nix extra) und Interface (DM 50,-, passend für Expansion-Stecker des Genie I/II, überflüssig bei TCS-FDC-Laufwerken)
 Software für Bildschirm-Hardcopy (Klötzchengraphik, inverse Zeichen und HRG für Genie I/II/III; bitte angeben!)
 Das Schriftbild und die Graphikmöglichkeiten sind euch aus meinen Artikeln sattem bekannt.
 Preisvorstellung: DM 500,- (VB)
 Arnulf Sopp

WER HAT GENIETEXT
 für GENIE III mit zugehörigem MIDABAS
 evtl. auch leihweise für einmalige Anwendung

UNIDAT
 ebenso ??????????????????
 Richard Rensch, Bahnhofstr. 100, 7128 LAUFFEN AM NECKAR

Software zum Nulltarif und Programme auf Probe

Die Preise für Computer und Lernprogramme, Rezepte, Flugsimulatoren, kurz, alles nur Wunschenswerte. Gegen Einsendung von 10 Mark erhält man einen kompletten Programm-katalog auf Diskette.

Ganz kostenlos ist die Software natürlich nicht: Je nach Bestellmenge wird pro Diskette eine Bearbeitungsgebühr berechnet. Und: Die Qualität der Programme ist unterschiedlich. Neben früheren Versionen heute kommerziell angebotener Programme sind da auch einfache BASIC-Spielchen im Angebot. Aber das ist eigentlich Nebensache, setzt man einmal die rund 15 Mark für eine Diskette mit 20 Programmen gegen die mindestens 40 bis 50 Mark, die man sonst für ein einziges Spielprogramm zahlen muß.

Eine weitere Alternative zur teuren kommerziellen Software ist unter der Bezeichnung „Shareware“ zu haben, die ebenfalls von EcoSoft vertrieben wird. Die Autoren geben ihre Programme zunächst kostenlos ab und überlassen es dem Anwender, sie auf Brauchbarkeit zu testen. Sagt dem das Programm zu, läßt er sich gegen Übersendung eines entsprechenden Geldbetrages beim Autor registrieren. Dafür erhält er die jeweils neueste Version, oft auch ausführliche Handbücher, und kann den Autor direkt um Hilfe bitten.

Die EcoSoft (Economy Software AG) in 7890 Waldshut-Tiengen, Kaiserstraße 21, bietet seit Jahresanfang Freiprogramme für die Systeme Commodore 64, Apple-Computer und IBM PC an.

So stehen für den populären C 64/128 etwa 160 Disketten mit rund 1500 Einzelprogrammen und für Apple-Computer sogar 250 Disketten mit insgesamt 6000 Einzelprogrammen zur Verfügung.

Die Preise für Computer und Lernprogramme, Rezepte, Flugsimulatoren, kurz, alles nur Wunschenswerte. Gegen Einsendung von 10 Mark erhält man einen kompletten Programm-katalog auf Diskette.

Ganz kostenlos ist die Software natürlich nicht: Je nach Bestellmenge wird pro Diskette eine Bearbeitungsgebühr berechnet. Und: Die Qualität der Programme ist unterschiedlich. Neben früheren Versionen heute kommerziell angebotener Programme sind da auch einfache BASIC-Spielchen im Angebot. Aber das ist eigentlich Nebensache, setzt man einmal die rund 15 Mark für eine Diskette mit 20 Programmen gegen die mindestens 40 bis 50 Mark, die man sonst für ein einziges Spielprogramm zahlen muß.

Eine weitere Alternative zur teuren kommerziellen Software ist unter der Bezeichnung „Shareware“ zu haben, die ebenfalls von EcoSoft vertrieben wird. Die Autoren geben ihre Programme zunächst kostenlos ab und überlassen es dem Anwender, sie auf Brauchbarkeit zu testen. Sagt dem das Programm zu, läßt er sich gegen Übersendung eines entsprechenden Geldbetrages beim Autor registrieren. Dafür erhält er die jeweils neueste Version, oft auch ausführliche Handbücher, und kann den Autor direkt um Hilfe bitten.

Bits übers Radio

Unter diesem Titel bringt der Computer-Laden einmal im Monat Wissenswertes, Aktuelles und Kurioses aus der Welt der großen und kleinen Computer. Die Sendung verrät Tips und Tricks für Anfänger. Profis und Amateure berichten, wie sie den Zugang zum Computer gefunden haben. Und schließlich schicken die Macher der Sendung sogar Programme über den Äther, die sich die Zuhörer aufnehmen und in ihren Computer laden können. Die Sendung wird jeweils an Dienstagen von 8.30...9.00 Uhr und von 14.30...15.00 Uhr ausgestrahlt (Südfunk, Stuttgart).

Folgende Termine sind vorgesehen:

Beginn 8.30 Uhr	Beginn 14.30 Uhr
25. November	18. November
23. Dezember	16. Dezember
3. Februar	27. Januar
3. März	24. Februar
31. März	24. März
5. Mai	28. April
2. Juni	26. Mai
30. Juni	23. Juni

49

Nun noch etwas in eigener Sache.
In den 10 Monaten, seit ich die PB verwalte, ist eine ganz
ansehnliche Menge Programm-Anforderungen bei mir eingegangen.
Da macht man schon mal lustige Erfahrungen! Z.B.: Irgendwann
irgendwo muß ich mal im Zusammenhang mit der Beschreibung der
persönlichen Computer-Konfiguration (zu Deutsch: Ausstattung
des Computers nebst Peripherie) erwähnt haben, daß zu meinem
alten TANDY (TR80/M1) ein "EXPANSIONS-INTERFACE" gehört.
Das müssen einige Kameraden total mixverstanden haben! (Kein
Tippfehler.) Einige glaubten offenbar, das sei eine "Dehn-
maschine", die ich erfunden hätte ("to expand" heißt ja in
der Tat "ausdehnen", "erweitern"; aber Expansion ist nicht
dasselbe wie Expander!). Diese gutgläubigen Optimisten baten
mich um Kopie von einigen Dutzend Programmen und schickten
mir dazu ein oder auch zwei Disketten, die auf 40 Spuren mit
SS/SD formatiert waren, wo (bekanntlich) etwa 78 freie Gra-
nules draufgehen, so daß ich davon ausgehen mußte, daß sie

27.

50

WDR-Files GRF Progr 0348 Teile: 18 / Grans: 167 Disk 12
 ===== f.Comp. Mod1,z.T.4 Quelle WDR-COMPUTERCLUB
 Verfasser versch. eingetippt von H. Obermann
 Grafikprogramme: APFELMANN/DAGON/DRIVEUM/HRGCHES/KREISE/
 SHUTTLE/SINXX/SURFACE/TRON/VERSCHLU/WDR2306/ZYKLOID/DRUCKBLD
 LIESBILD/SCANBILD.-Ueberblick in: PROGR/TXT

CALL/CMD UTL Progr 0349 Teile: 02 / Grans: 02 Disk 12
 ===== f.Comp. alle Quelle
 Verfasser H.Obermann eingetippt von H.Obermann
 Maschinenroutinen von DOS aus "callen".
 Z.B.: CALL 01C9 --> CLS

EXECUTE/CMD UTL Progr 0350 Teile: 02 / Grans: 03 Disk 12
 ===== f.Comp. alle Quelle
 Verfasser Hartmut Obermann eingetippt von Hartm. Obermann
 Das Progr. erlaubt es, kleine von Hand assemblierte Maschi-
 nen-Routinen direkt einzugeben und auszufuehren. Z.B.:
 210000110070 entspr.LD HL,0000h LD DE 7000h

LIESHRG/BAS GRF Progr 0351 Teile: 01 / Grans: 01 Disk 12
 ===== f.Comp. Mod. 1/4 Quelle
 Verfasser Hartmut Obermann eingetippt von Hartm.Obermann
 Programm zum Einlesen von Modell-HRG-Files in die Model4-HRG

LPRINT/CMD UTL Progr 0352 Teile: 02 / Grans: 02 Disk 12
 ===== f.Comp. alle Quelle
 Verfasser Hartmut Obermann eingetippt von Hartm. Obermann
 erlaubt die direkte Ausgabe von Steuersequenzen vom DOS aus
 an einen Drucker.- Z.B.: LPRINT 1B40 ==> ESC \$
 (Initialisierungssequenz f. EPSON-Drucker)

OUT/CMD UTL Progr 0353 Teile: 02 / Grans: 02 Disk 12
 ===== f.Comp. alle Quelle
 Verfasser Hartmut Obermann eingetippt von Hartm. Obermann
 erlaubt die Ausgabe von HEX-Werten an Ports direkt im DOS.
 Z.B.: OUT 01,00 ==> "HRG einschalten!" (Model 1)

POKE/CMD UTL Progr 0354 Teile: 02 / Grans: 02 Disk 12
 ===== f.Comp. alle Quelle
 Verfasser Hartmut Obermann eingetippt von Hartm. Obermann
 wie BASIC-Befehl 'POKE', jedoch unter DOS verwendbar. Z.Bsp.
 POKE 4210,60 ==> "Double speed!" (Model 4)

GRAFLIX/BAS GRF Progr 0355 Teile: 07 / Grans: 15 Disk 12
 ===== f.Comp. alle Quelle COMPERS.3.10.84
 Verfasser F.DEBATIN eingetippt von Hartm. Obermann
 Gutes Grafikprogramm mit eigenem HRG-Treiber "HRGLIX/CMD".
 Grafik kann mit Text aus DOTWRITER-Zeichen gemischt werden.
 Start: HIMEM CFOOH HRGLIX RUN"GRAFLIX/BAS"

GRAPE21 GRF Progr 0356 Teile: 18 / Grans: 65 Disk 10
 ===== f.Comp. alle Quelle M.WINTER,STUTTG.
 Verfasser Martin Winter eingesandt von R. Jablotschkin.
 Das bekannte Programm fuer die hochaufloesende Grafik wurde
 von M.Winter fuer den CLUB80 freigegeben. Es darf nur inner=
 halb des Clubs weitergegeben werden!

TRSSYS/SRC UTL Progr 0357 Teile: 01 / Grans: 06 Disk 13
 ===== f.Comp. TRS 80 Quelle
 Verfasser SCHROEDER, Gerald eingetippt von dto.
 Laedt alle Systemfiles in zwei Banks. Man kann somit die
 Systemdiskette aus Laufwerk 0 entfernen.
 (Arbeitet mit Helmut Bernhards Banker.)

SYSCOPY1/SRC UTL Progr 0358 Teile: 01 / Grans: 06 Disk 13
 ===== f.Comp. GENIE IIs Quelle
 Verfasser SCHROEDER, Gerald eingetippt von dto.
 Arbeitet aehnlich wie TRSSYS/SRC (Nr.357), jedoch
 n u r a u f G E N I E I I s !

TRSRAM/SRC UTL Progr 0359 Teile: 01 / Grans: 11 Disk 13
 ===== f.Comp. TRS 80 Quelle
 Verfasser SCHROEDER, Gerald eingetippt von dto.
 initialisiert RAMDISK als Lw2 oder 3 mit 64-896KB Speicherpl
 Parameter: F=RAMDISK formatieren; 1-E: RAMDISK mit 1-14*64K.
 N=RAMDISK entfernen, alt.Zust.restaurieren.

RAMDISK/SRC UTL Progr 0360 Teile: 01 / Grans: 10 Disk 13
 ===== f.Comp. GENIE IIs Quelle
 Verfasser SCHROEDER, Gerald eingetippt von dto.
 Wie "TRSRAM/SRC" (Progr.-Nr.359), jedoch
 n u r f u e r G E N I E I I s !

COMPARE/SRC UTL Progr 0361 Teile: 01 / Grans: 09 Disk 13
 ===== f.Comp. Quelle
 Verfasser SCHROEDER, Gerald eingetippt von dto.
 Vergleicht zwei CMD-Files und beruecksichtigt dabei unter=
 schiedliche Laengen oder Spruenge bei den Ladeadressen.

SYS8/SRC UTL Progr 0362 Teile: 01 / Grans: 12 Disk 13
 ===== f.Comp. Quelle
 Verfasser SCHROEDER, Gerald eingetippt von dto.
 Ein neues SYS8/SYS fuer 6-DOS. Par.\$ fehlt.Neu: xxxxxxxx/yyy
 x/y=Buchst./Ziffer, "?"=Zeichen beliebig, "*"="alle folgenden
 Zchn.belieb.-Nur entspr.Files werden angez.

TRSMAC/CMD UTL Progr 0363 Teile: 02 / Grans: 16 Disk 13
 ===== f.Comp. Quelle MIT SOURCE-CODE
 Verfasser SCHROEDER, Gerald eingetippt von dto.
 Hilfsprogr.f.Eingaben. Wird ueb.Parametersatz aufgerufen. Es
 koennen dann Texteingaben gemacht oder Flags umgeschaltet
 werden. Ein spezieller "Full-Screen-Editor"

TRSTST/CMD UTL Progr 0364 Teile: 02 / Grans: 18 Disk 13
 ===== f.Comp. Quelle
 Verfasser SCHROEDER, Gerald eingetippt von dto.
 Fuehrt das Programm "TRSMAC/CMD" (Nr.363) vor. Mit Source-C.

TRSDemo/BAS UTL Progr 0365 Teile: 01 / Grans: 02 Disk 13
 ===== f.Comp. Quelle
 Verfasser SCHROEDER, Gerald eingetippt von dto.
 Fuehrt ebenfalls das Programm "TRSMAC/CMD" (Nr.363) vor.

Programm-Eingaenge seit INFO 14 (ohne Publ.Dom.)-Stand: 22.10.86

TRSDEF/BAS UTL Progr 0366 Teile: 01 / Grans: 15 Disk 13
===== f.Comp. Quelle
Verfasser SCHROEDER, Gerald eingetippt von dto.
Zum Erstellen von Eingabemasken f. BASIC-Programme.
Siehe hierzu auch das Hilfsprogramm "BILDUMW/BAS" (Nr.367).

BILDUMW/BAS UTL Progr 0367 Teile: 01 / Grans: 03 Disk 13
===== f.Comp. Quelle
Verfasser SCHROEDER, Gerald eingetippt von dto.
Hilfsprogramm zu "TRSDEF/BAS" (Nr.366). Wandelt die Eingabe-
maske in BASIC-Anweisungen um (auch fuer andere Programme
verwendbar).



Wer möchte Diskothekar werden? =====

Wenn dieses Jahr zu Ende geht, werde ich die Programmbibliothek unseres CLUB80 genau ein Jahr lang verwaltet haben!

Es war eine interessante Arbeit. Ich hatte nicht nur das beruhigende Gefühl, außer dem pflichtgemäßen Clubbeitrag in DM auch einen physikalischen und physiologischen Beitrag (Tasten Tippen und Gehirnzellen Strippen) zu leisten, sondern ich habe auch viel dabei zugelernt. Von dem Programm-Reichtum, der mir auf diese Weise unverlangt (und unverdient) zugeflossen ist, ganz abgesehen.

Nun hat sich für mich aber eine neue Situation ergeben. Aus sehr persönlichen Gründen muß ich dieses schöne Hobby jetzt leider aufgeben.

Clubkollegen - vor allem diejenigen, von denen man sonst noch nie gehört hat - haben hier eine Chance, sich zu Hause in aller Gemütlichkeit um den Club verdient zu machen, wenn sie ansonsten keine Lust, Übung, Mut oder sonstwasweißich haben, selbst Programme zu schreiben und in unserem INFO zu veröffentlichen!

Wer also Lust hat, der melde sich bald bei mir! Ihm steht nicht nur ein Potential von nahezu 600 Programmen zur Verfügung (auf nur 40 Disketten) nebst Anleitungen, sondern es eröffnet sich auch eine Fülle interessanter Kontakte, auf die man sonst gar nicht gekommen wäre.

Die Einarbeitung ist einfach. Ich habe eine für kluge Köpfe (ihre Stückzahl geht aus der Mitgliederliste hervor) völlig hinreichende Anleitung - nur 1 Seite! - für die Datenbank SUPER ausgearbeitet (die ausführliche steht natürlich auch zur Verfügung). Viele von euch kennen sie. Sie ist zwar etwas langsam (BASIC), hat aber den Vorteil, sehr sicher im Umgang und flexibel genug zu sein, um sich vielen Bedürfnissen anzupassen; z.B. Ausdrucken mit beliebiger Formatierung und Felder-Auswahl, selektive Suchfunktion mit mehreren Prioritätsstufen, wobei binäres Suchen sehr schnell geht, usw. Gerade weil sie in BASIC vorliegt, läßt sie sich leicht nach eigenem Wunsch und Geschmack verändern. So habe ich die Form, wie die "Diskothek" (Programmbibliothek; siehe in diesem Heft) ausgedruckt wird, durch einfache Eingriffe in das Programm selbst gestaltet und natürlich an meinen Drucker (EPSON) angepaßt - was jederzeit leicht geändert werden kann.

Fantastische Hilfe beim Kopieren leistet die Utility (wirklich eine "Nützlichkeit"!) DIS/CMD! Wird ein Programm gewünscht, lege ich nur die Diskette mit der vom Besteller genannten Nummer ein, sage "S Filename" (=Suche Programmnamen); schon springt der Cursor dorthin - falls das Programm drauf ist; andernfalls hat der Besteller eine falsche Disk-Nummer genannt. Genau dann hilft die Datei SUPER weiter. Sonst, wenn die Disknummer stimmt, wird SUPER überhaupt nicht gebraucht. Ich habe es in den 10 Monaten bisher ein einziges Mal gebraucht! (Am 22.10.86. Der Gute, der die WDR-Files bestellte, hatte mir überhaupt keine Disknummer genannt!) SUPER ist dann mein NOKIXEL <bitte rückwärts lesen!> Aber auch nur dann! Der Cursor springt also sofort auf das gewünschte File. Dann tippe ich nur noch "C n" - fertig; und DIS/CMD kopiert mir das Programm auf Laufwerk n mit der Diskette des "Kunden"! Keine Eingabe von "COPY" und Filenamen usw.usf.

Für diejenigen, die die Mitgliederliste mit den Anschriften sorgfältig und gewissenhaft abgetrennt haben, damit sie unser INFO auch einmal einem TRS80iger, GENIELER oder KOMTEKER, der noch 'außen vor' steht, zeigen und für uns werben können, wiederhole ich meine Anschrift hier nochmal:

K.-J.Mühlenbein, Am Mönchgarten 28, 6940 Weinheim

Ich rechne mit zahlreichen Bewerbern! Dann entscheiden: Ver- und Vorstand. Bis bald also! Euer

K.J.

1. Vorsitzende

Peter STEVENS
Postfach 34
4600 Dortmund 1
VBN 0231 / 373883

2. Vorsitzende

Hartmut OBERMANN
Schwalbacher Straße 4
6200 Heidenrod 1
VBN 06124 / 3913

Hardwarekoordinator

Eckehard KUHN
Im Dorf 14
/443 Frickenhausen 1
VBN 0722 / 43417

Diskotheke

Klaus-Jürgen MÜHLENBEIN
Am Mönchgarten 28
6940 Weinheim -LÜ.
VBN 06201 / 30852

Redaktion

Jens NEUEDER
Panoramastraße 21
/178 Michelbach /Bilz
VBN 0791 / 42811

Autoren

Helmut Bernhardt	X	Dieter Kasper
Eckehard Kuhn	X	Klaus-J. Mühlbein
Kurt Müller	X	Jens Neuder
Hartmut Obermann	X	Gerald Schröder
Annulf Sopp	X	

sowie Artikel aus: 88 Micro

Bankverbindung des CLUB 88

Postgirokonto Peter STEVENS
Sonderkonto CLUB 88
Konto-Nummer 285 491 - 465
Postgiroamt Dortmund
BLZ 440 100 46

Das INFO erscheint zweimonatlich.

Es erfolgt keine Zensur oder Kontrolle
der jeweiligen eingeschickten Infobeiträge
durch die Redaktion.

Hallo Club-88er,

leider kann ich Euch jetzt erst das neueste INFO bieten. Dies hatte private sowohl auch berufliche Gründe. Weiterhin hat es im Club eine Umorganisation gegeben, die Euch natürlich auch bekanntgegeben werden musste. Darum leider erst jetzt das 16. Clubinfo.

Da dies auch das letzte INFO sein wird, daß Ihr in diesem Jahr bekommt (das Nächste erscheint Anfang/Mitte Januar 87), möchte ich mich bei allen Mitwirkenden, an unserer Clubinfo, recht herzlich für die geleistete Arbeit bedanken.

Gleichzeitig geht aber auch meine Bitte an alle, auch im nächsten Jahr wieder so produktiv zu sein. Auch Autoren-Neulinge sind im Kreis der INFO-Schreiber herzlichst willkommen.

Durch die INFO wird doch eine Menge an Informationen an die Clubmitglieder gebracht. Sicherlich war für jeden von Euch etwas Brauchbares dabei. Ich denke auch, daß wir mit dem Konzept unserer Clubinfo recht gut liegen (Zumindest gab es keine größeren Klagen). Bedenkt man dabei noch, daß wir mehr und mehr eine **RARITÄT** -mit unserem 8-Bit'er- besitzen und auf uns alleine gestellt sind, ist die Aktivität in unserem Club doch recht gut. Ich wünsch Euch und mir, daß es auch weiterhin so bleibt.

Am INFO-Ende sind diesmal zwei **SONDERHEFTE** beigelegt. Ich habe mich für diesen Weg entschieden, da die behandelten Themen recht umfangreich sind. Gleichzeitig möchte ich damit auch die Arbeit der Autoren würdigen, die sich wieder einmal sehr viel Mühe gemacht haben.

Als Vorstandsmitglied möchte ich Euch **ZUM SCHLUSS MEINES SCHLUSSES** noch an die anstehenden Termine erinnern. Bitte merkt Euch die wichtigsten Termine vor.

Einstweilen wünsch Euch viel Spaß an der neuesten INFO, ein **FROHES FEST** und einen **GUTEN RUTSCH IN'S NEUE** und Allzeit **HAPPY HACKING**.

Euer

Jens Neuder

Name	Vorname	Straße	PLZ Stadt	Telefon privat	// geschäftlich
Albers	Herbert	Zum Düwelshöpen 14	2117 Wistedt	04182 /8799	// -
Beckhausen	Wolfgang	Vuerfelser-Kaule 30	5060 Bergisch-Gladbach 1	02204 /62781	//
Bernhardt	Helmut	Hafenstraße 7	2305 Heikendorf	0431 /241987	// 0431 /74047
Buskowiak	Thomas	Eschersheimer Landstr. 257	6000 Frankfurt 1	069 /5601621	//
Böcker	Dieter	Lehmweg 4	2930 Varrel 1	04451 /7640	//
Böckling	Ulrich	Am Sonnenhang 11	5414 Vallendar	0261 /69522	// 02631 /895168
Dreyer	Gerald	Am Speiergarten 8	6200 Wiesbaden-Bierstadt	06121 /508218	// -
Drowälder	Bernd	Hügel 1	4441 Mettringen	05233 /4320	// 05255 /1236
Emmrich	Helmut	Waldstraße 5	6682 Ottweiler	06824 /4114	// -
Frink	Thomas	An der Schleifmühle 2	5042 Erftstadt	02235 /6255	//
Grajewski	Werner	Zedernweg 29	4220 Dinslaken	02134 /54573	//
Hartel	Eberhard	Neenstetter Straße 20	7901 Breitingen	07340 /7281	//
Heile	Heinz-Dieter	Blankensteiner Straße 13	4320 Hattingen	02324 /20458	//
Held	Manfred	Stirnerstraße 22	8835 Pleinfeld	09144 /6563	// 0911 /2195245
Hermann	Klaus	Gartenstr. 22	7401 Pliezhausen	07127 /70024	//
Hill	Peter	Eckstraße 36	6750 Kaiserslautern 31	0631 /54782	//
Hummel	Anton	Schubertstr. 2	7612 Haslach	07832 /8289	//
Jablotschkin	Rainer	Thiekamp 29	4780 Lippstadt 8	02948 /542	// 02921 /70431
Kasper	Dieter	Zeppelinstr. 9	8952 Marktoberdorf	08342 /1630	// 089 /522071
Konrad	Josef	Anzengruberstraße 35	8038 Gröbenzell	08142 /8494	// -
Kopschina	Peter	Strandallee 138	2409 Scharbeutz	-	// -
Krüger	Karl-Herbert	Bruchweg 65	4920 Lemgo	-	// -
Kuhn	Eckehard	Im Dorf 14	7443 Frickenhausen 1	07022 /45417	// 07022 /77442
May	Holger	Marienstr. 9	5768 Sundern 2	02935 /1668	// -
Misioch	Waldemar	Adenauerring 25	8505 Röthenbach a. d. Pegnitz	0911 /506051	// 0911 /668151
Mühlenbein	Klaus-Jürgen	Am Mönchgarten 28	6940 Weinheim -Lützel-sachsen	06201 /55052	// -
Müller	Kurt	Soltstraße 24a	2050 Hamburg 80	040 /7246883	// 04103 /702662
Neueder	Jens	Panoramastraße 21	7178 Michelbach /Bilz	0791 /42877	// 0791 /44-667
Obermann	Hartmut	Schwalbacher Str. 6	6209 Heidenrod 1	06124 /3913	// -
Perschbach	Patrick	Waldstr. 52	5000 Köln 91	0221 /872118	//
Piller	Walter	Rohnenstraße 8	CH-8835 Feusisberg	01 /7847418	//
Raggan	Hans	Backnanger Weg 36	7146 Tamm	07141 /603611	// 0711 /2630473
Rank	Heinrich	Frühlingstraße 2	8080 Fürstenfeldbruck	08141 /3791	//
Rensch	Richard	Bahnhofstraße 100 (Postf. 226)	7128 Lauffen am Neckar	07133 /4167	// 07133 /8415
Retzlaff	Bernd	Kleiner Sand 98	2082 Uetersen	04122 /43551	// 04103 /7025310
Rychlik	Andreas	Königsberger Allee 120	4100 Duisburg 1	0203 /331383	// 0203 /331383
Schmitz	Paul-Jürgen	Lübeskerstraße 6	6236 Eschborn	-	// -
Schneider	Manfred	Rheinkasseler Weg 11	5000 Köln /1	0221 /707044	//
Schrewe	Christian	Fliederweg 32	4000 Düsseldorf 31	0203 /740897	//
Schröder	Gerald	Am Schützenplatz 14	2105 Seevetal 1	04105 /2602	// -
Schäfer	Walter	Rathausstr. 4	8160 Miesbach	08025 /1631	// 08025 /41247
Smerling	Frank	Tangstedter Str. 5	2080 Pinneberg	04101 /207284	//
Sopp	Arnulf	Wakenitzstr. 8	2400 Lübeck 1	0451 /791926	// -
Spieß	Peter	Trugenhofenerstraße 27	8859 Rennertshofen 1	08434 /454	// 08431 /7041684
Stephan	Hans-Martin	Am Glasesch 9a (Postf. 1207)	4506 Hagen a.TW.	05401 /99585	// 05401 /90037
Stevens	Peter	Postfach 56	4600 Dortmund 1	0231 /593883	// 0231 /593883
Trapp	Harald	Kranichstr. 46	4270 Dorsten 1	02362 /42497	// 02362 /23127
Troesch	Eberhard	Altenessener Str. 414	4300 Essen 12	0201 /342324	//
Volz	Uliver	Dusestraße 13	7000 Stuttgart 80	0711 /731285	//
Wagner	Günther	Gartenstraße 4	8201 Neubuvern	08035 /3361	// -
Weiß	Dieter	Bürglestraße 3	7209 Wehingen	07426 /7194	//
Mucherer	Jürgen	Menzelstraße 1	7750 Konstanz	07531 /54686	// -
Zwickel	Walter	Lengfelden 123	A- 5101 Bergheim	0043662/51130	// -